LÜERS

CPC 464 BASIC-PROGRAMME

EIN DATA BECKER BUCH

LÜERS

CPC 464

BASIC-PROGRAMME

EIN DATA BECKER BUCH

ISBN 3-89011-049-5

Copyright (C) 1984 DATA BECKER GmbH
Merowingerstr. 30
4000 Düsseldorf

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der DATA BECKER GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Wichtiger Hinweis!

Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technische Angaben und Programme in diesem Buch wurden von den Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler sind die Autoren jederzeit dankbar.

INHALTSVERZEICHNIS

Sehr geehrte, liebe Leserin!
Sehr geehrter, lieber Leser!
Sehr geehrte, liebe Programmiererin!
Sehr geehrter, lieber Programmierer!

Sie halten das Buch 'BASIC Programme' in Händen und wollen sich vielleicht erst einmal ein schnelles Urteil über dieses Buch erlauben. Dazu haben Sie sicherlich einen Blick ins Inhaltsverzeichnis geworfen ... gut so ... und nun wollen Sie noch schnell die Anrede überfliegen ... und dann geht es ab ins stille Kämmerlein zum Programmieren bzw. zum Programm eintippen in den Schneider CPC464!?

Lassen Sie mich bitte trotzdem noch für ein paar Worte zur Sprache kommen, bevor es richtig losgeht.

Hinter dieser Programmsammlung steckt ein anderes Konzept als man vielleicht auf den ersten Blick ahnen könnte. Diese Programmsammlung besteht nämlich nicht nur aus Programmen, nein auch aus über vierzig Seiten Text. Nicht daß mir die Programme ausgegangen wären oder mir die Ideen fehlten (würde ich gleich alles veröffentlichen, hätte dieses Buch knapp 400 Seiten!), nein, Sie sollen meiner Meinung nach nicht nur einfach tippen, tippen und nochmals tippen ... nein, Sie sollen auch etwas über Ihren CPC dazulernen.

Zwar ist nicht Programmzeile für Zeile genauestens erklärt, dafür fehlt der Platz und vielleicht auch Ihre Ausdauer. Aber auf interessante Details wird im Vortext hingewiesen, und zum großen Teil sind die Programme ja auch noch mit REM-Zeilen zur Erläuterng angefüllt.

Auch inhaltlich liegt der Schwerpunkt mehr auf der nützlichen Anwendungsseite, sei es, daß Sie Musik, Grafik oder Text editieren wollen, sei es, daß Sie planen können, Ihre Schallplatten oder die Bundesligaergebnisse gemeinsam mit dem CPC zu verwalten. Schließlich liegt noch ein weiterer Schwerpunkt in dieser Programmsammlung: Sie sollen ein klein wenig davon einen Eindruck bekommen, wie Ihr CPC 'innerlich' arbeitet. Programme wie

'Disassembler' und 'Variablenrefernzliste' sind nur zwei Beispiele dafür.

Auf jeden Fall, und lassen Sie mich damit schließen, sollen Sie Spaß daran verspüren, in die geheimnisvolle Welt der Computer und des CPC im besonderen immer tiefer einzudringen und vielleicht sogar durch die Programmsammlung selbst zum kleinen Programmierer aufzusteigen.

Noch einmal: Viel Freude und viel Spaß an der Arbeit mit diesem DATA BECKER-Buch wünscht Ihnen der Autor von Programmen und Text.

Speicher 1

Schauen wir uns das Resultat dieses Programms am Bildschirm an, so stellen wir fest, daß der Speicher, dessen Inhalt wir mit dem Befehl 'PEEK' abfragen nicht überall leer = '0' ist.

Zwar erscheinen auf dem Bildschirm nur anscheinend wirre Zahlen, jedoch werden wir gleich feststellen, daß sich hinter diesen Zahlen zum Teil sinnvolle Buchstaben und Zeichen verbergen. Hierzu ein Experiment:

Geben Sie folgendes in Ihren CPC ein: 'PRINT ASC("!")' (ENTER)
Ergebnis: 33. Das heißt: Hinter dem Ausrufungszeichen verbirgt sich für den Computer die Zahl 33. Probieren wir es andersherum:
Geben Sie ein: 'PRINT CHR$(33)' (ENTER)
Ergebnis: '!'. Damit haben wir den Computer dazu aufgefordert, die Zahl 33 wieder in ein Zeichen umzuwandeln. Das Ergebnis von Zahl 33 ist unser Ausrufungszeichen!

So können wir nun auch unser Zahlenmeer aus Programm Speicher 1 zu deuten lernen. Geben Sie zur Überprüfung gleich anschließend bitte das Programm 'Speicher 2' ein.

Hinweis zum Gebrauch von Programm 'Speicher 1':

Anfangs- und Endspeicheradresse sollten nicht weiter als 150 Zeichen auseinanderliegen; so bekommen wir in der Darstellungsweise alle gewünschten Daten auf eine Bildschirmseite.

```
10 REM Untersuchung des Speichers 1
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
50 MODE 1
60 REM Darstellungsraum eingrenzen
70 INPUT "Anfangsadresse eingeben ";a
80 INPUT "Endadresse eingeben ";e
90 IF e<a OR e>65535 OR a<0 OR a>65535 T
HEN GOSUB 210:GOTO 50
100 PRINT
110 REM Ausgabe des Speicherinhalts im
    eingegrenzten Darstellungsraum durch
    Zahlen
120 FOR n=a TO e
130 PRINT PEEK(n);
140 NEXT n
150 PRINT
160 PRINT
170 INPUT "Wollen Sie weitere Teile des
           Speichers untersuchen ( /N) "
;f$
180 f$=UPPER$(f$)
190 IF f$<>"N" THEN GOTO 10
200 END
210 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
220 PEN normal:GOSUB 230:RETURN
230 PRINT:PRINT TAB(7) "<Bitte eine Tast
e druecken>"
240 f$=INKEY$:IF f$="" THEN GOTO 240
250 RETURN
```

```
Anfangsadresse eingeben ? 1000
Endadresse eingeben ? 1150

 32  3  60  0  230  242  34  78  34  32
 235  32  160  32  29  111  1  0  6  0
 200  0  152  0  45  0  210  0  187  32
 13  0  0  101  102  102  101  107  244
 1  191  1  191  32  234  40  25  12
 41  32  34  70  97  108  115  99  104
 101  32  69  105  110  103  97  98
 101  33  34  0  24  0  220  0  187  32
 13  0  0  110  111  114  109  97  236
 1  159  32  30  230  0  1  201  0  43
 0  230  0  191  1  191  32  234  40
 21  41  32  34  60  66  105  116  116
 101  32  101  105  110  101  32  84
 97  115  116  101  32  100  114  117
 101  99  107  101  110  62  34  0  30
 0  240  0  3  0  0  230  239  255  67
 1  161  32  3

Wollen Sie weitere Teile des
Speichers untersuchen ( /N) ? █
```

Speicher 2
==========

Nachdem Sie dieses Programm abgetippt und es mit 'RUN' gestartet haben passiert folgendes: Was auch immer Sie tun, entweder passiert überhaupt nichts oder es passiert etwas Unvorhergesehenes (der Bildschirm wird gelöscht, der Mode wird verändert, PEN und PAPER ändern sich ...). Warum dies?

Wie schon in anderen Programmen dieser Sammlung aufgeführt: Es gibt sinnvoll direkt darstellbare Zeichen ('CHR$(32) = Leerzeichen bis 'CHR$(126)' = Schlangenlinie bzw. bis 'CHR$(255)' = Grafikzeichen), es gibt aber auch andere CHR$-Zeichen, die sich nicht so eindeutig im Bild festhalten lassen ('CHR$(2)' = Textcursor ausschalten bzw. 'CHR$(7)' = Klingel ertönen lassen - Einzelheiten siehe im Handbuch Kapitel 9).

Also müssen wir noch ein drittes Programm schreiben, um nur die Zahlen als Zeichen auf dem Bildschirm entsprechend darzustellen, die in diesem Zusammenhang auch sinnvoll darstellbar sind. Dies geschieht nun mit einigen Programmerweiterungen in Programm 'Speicher 3'.

Hinweis zum Gebrauch von Programm 'Speicher 2':

Wenn auf dem Bildschirm nun nichts rechtes mehr zu erkennen ist, Sie aber das eingegebene Programm weiterbenutzen wollen, so heißt es nun tüfteln: Schlagen Sie Ihr Handbuch Kapitel 9 auf und versuchen Sie die entsprechenden Störfaktoren durch Eingabe von 'PRINT' und dem entsprechenden 'CHR$('-zeichen zu loeschen.

```
10 REM Untersuchung des Speichers 2
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                        Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
50 MODE 1
60 REM Darstellungsraum eingrenzen
70 INPUT "Anfangsadresse eingeben ";a
80 INPUT "Endadresse eingeben ";e
90 IF e<a OR e>65535 OR a<0 OR a>65535 T
HEN GOSUB 210:GOTO 50
100 PRINT
110 REM Ausgabe des Speicherinhalts im
    eingegrenzten Darstellungsraum durch
    Zahlen
120 FOR n=a TO e
130 PRINT CHR$(PEEK(n));
140 NEXT n
150 PRINT
160 PRINT
170 INPUT "Wollen Sie weitere Teile des
           Speichers untersuchen ( /N) "
;f$
180 f$=UPPER$(f$)
190 IF f$<>"N" THEN GOTO 10
200 END
210 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
220 PEN normal:GOSUB 230:RETURN
230 PRINT:PRINT TAB(7) "<Bitte eine Tast
e druecken>"
240 f$=INKEY$:IF f$="" THEN GOTO 240
250 RETURN
```

Speicher 3

Mit Eingabe dieses Programms werden gleich zwei Fliegen mit einer Klappe geschlagen: Einerseits erscheint die Bildschirmwiedergabe des Speicherinhalts nun geordneter (jeweils Anfangsadresse, Inhalte der Anfangsadresse und der folgenden sieben Bytes sowie CHR$-Ausdruck der acht Bytes, alles zusammen in einer Bildschirmzeile), außerdem werden jetzt nur noch die sinnvollen 'CHR$(' dargestellt (zwischen 32 = Leerzeichen und 126 = Schlangenlinie). Für 'nicht ohne weiteres sichtbare CHR$' haben wir einfach 'CHR$(46)' = '.' zur Darstellung herangezogen.

Sie können wählen: entweder geben Sie die Anfangsadresse ein und drücken anschließend lediglich die (ENTER)-Taste; so werden nur die Anfangsadresse und die Inhalte der entsprechenden acht Bytes samt CHR$-Darstellung ausgedruckt oder um die Speicherinhalte auch über eine Zeile hinaus abbilden zu können, gilt es nach dem erstmaligen Drücken der (ENTER)-Taste sofort anschließend (während des Bildschirmaufbaus) irgendeine andere Taste mit Wiederholfunktion (z.B. die Leertaste) andauernd niederzudrücken. So erscheinen zu je acht Bytes pro Zeile die sich anschließenden Speicherinhalte auf dem Monitor.

Schauen wir uns doch einmal an, wo unser BASIC-Programm 'Speicher 3' geblieben ist. Beim CPC wie auch bei jedem anderen Computer werden Befehle zum Teil nur als einzelne Zahlen = Token abgespeichert, hingegen bleiben Worte, die hinter REM-Anweisungen stehen, vollkommen erhalten. Geben Sie z.B. als Anfangsadresse '1008' ein ... rechts im CHR$-Bereich müßten Sie nun das Wörtchen 'jeweils' entdecken ... schauen wir in das Programmlisting -) siehe Zeile 140:

```
'140 REM Die Bildschirmzeile wird mit
         jeweils acht Bytes gefuellt'
```

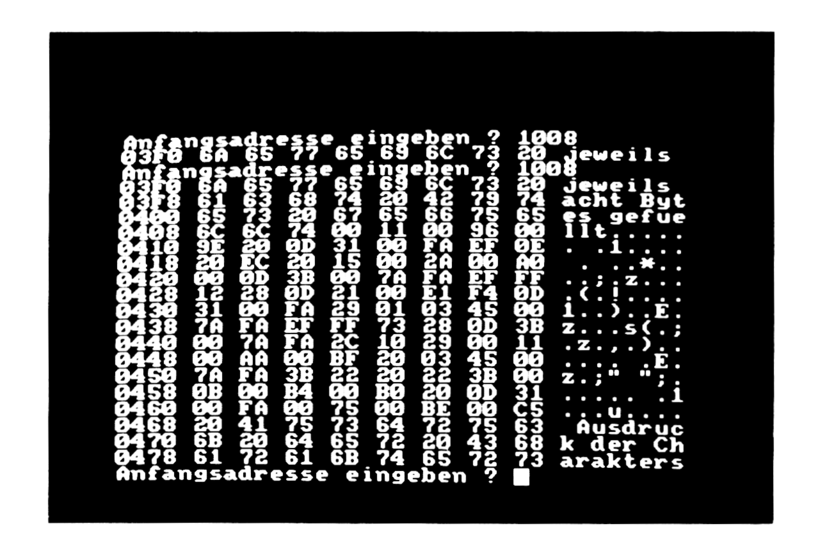
... das funktioniert allerdings nur, wenn Sie
das Programm Zeile für Zeile genau so wie im Buch abgetippt haben, ohne auch nur ein Leerzeichen vergessen zu haben!

Geben wir '1008' als Anfangsadresse ein und drücken unverzüglich eine Taste mit Wiederholfunktion z.B. die Leertaste, so werden uns noch weitere Bestandteile aus unserem Programm 'Speicher 3' begegnen.

Hinweis zum Gebrauch des Programms 'Speicher 3':

Wie Sie sicherlich bemerkt haben, wird hier als Zahlsystem nicht das uns vertraute Dezimalsystem (Zahlen aus 0 bis 9 zusammengesetzt) sondern das Hexadezimalsystem (Zahlen aus 0 bis F zusammengestzt) angewandt. Dies hat folgende Gründe: Einerseits ist das Hexadezimalsystem 'das' System der Hacker und Maschinensprachfreaks ... so wollen wir es hier als 'Speicherdurchforster', was gleichbedeutend mit 'Hacker sein soll, auch anwenden! Andererseits läßt sich auf 40 Spalten mit dem Dezimalsystem nicht so viel Information unterbringen wie mit Hilfe des Hexadezimalsystems ('255' Dez. = 'FF' Hex. bzw. '65535' Dez. = 'FFFF' Hex.).

Dezimalumwandlung in Hex läßt sich mit dem CPC sehr leicht mit dem Befehl 'HEX$(' in die Tat umsetzen z.B. 'PRINT HEX$(65535)', Ergebnis = 'FFFF'.



```
10 REM Untersuchung des Speichers 3
20 REM Untersuchung des Speichers mit
       PEEK und CHR$( in uebersichtli-
       cher Form
30 REM CPC464 Basic Programme
40 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
60 KEY DEF 18,1
70 MODE 1
80 REM Darstellungsraum eingrenzen
90 INPUT "Anfangsadresse eingeben ";a
100 IF a<0 OR a>65535 THEN GOSUB 330:GOT
O 70
110 a$=HEX$(a,4)
120 REM Die naechsten Zeilen helfen,
        einen optimalen Bildschirmaus-
        druck mit 40 Zeichen/Zeile zu
        ermoeglichen
130 PRINT a$;" ";
140 REM Die Bildschirmzeile wird mit
        jeweils acht Bytes gefuellt
150 FOR z=0 TO 7
160 zz=PEEK(a+z):zz$=HEX$(zz,2)
170 PRINT zz$;" ";
180 NEXT z
190 REM Ausdruck der Charakterstrings
        bei besonderer Beachtung, wenn
        der PEEKwert <32 bzw. >126 ist
200 FOR z=0 TO 7
210 zz=a+z
220 zz=PEEK(zz)
230 IF zz<32 THEN zz=46
240 IF zz>126 THEN zz=46
250 PRINT CHR$(zz);
260 NEXT z
270 PRINT
280 a=a+8
```

```
290 a$=INKEY$
300 IF a$="" THEN GOTO 90
310 GOTO 110
320 END
330 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
340 PEN normal:GOSUB 350:RETURN
350 PRINT:PRINT TAB(7) "<Bitte eine Tast
e druecken>"
360 f$=INKEY$:IF f$="" THEN GOTO 360
370 RETURN
```

Speicher 4

Bisher haben wir den Speicher durchsucht und dabei zum Teil nur Zahlen (Speicher 1), wirre Zeichen (Speicher 2) oder aber gar schließlich beides in geordneter Form auf dem Bildschirm erzeugt (Speicher 3). Beim Listing von 'Speicher 3' konnte im Vergleich der linken zur rechten Bildschirmseite deutlich werden, daß wahrlich unser BASIC-Programm nur aus Zahlen besteht, die erst im Nachhinein wieder vom CPC in Buchstaben und Zeichen umgewandelt wurden.

Hierzu muß man wissen, wie der Computer Daten und Programme abspeichert: Um Speicherplatz zu sparen, werden BASIC-Befehle nach der Eingabe durch (ENTER) in Abkürzungen umgewandelt, die sogenannten Tokens (siehe Anhang). So verbraucht der Befehl 'PRINT' nicht fünf Bytes (da fünf Buchstaben), sondern lediglich ein Byte.

Schreiben wir jedoch in eine REM-Zeile 'PRINT' oder geben in eine Programmzeile 'PRINT "PRINT"' ein, so wird das in diesem Fall nicht als Befehl zu interpretierende 'PRINT' Buchstabe für Buchstabe abgespeichert, d.h. hierzu werden wie zu erwarten fünf Speicherplätze = fünf Bytes in unserem CPC benötigt.

Suchen wir nach dem Vorhandensein irgendwelcher ausgeschriebenen Wörter im BASIC-Programm, so kann dies sehr lange dauern, bei 65535 Speicherplätzen ähnlich einer Suche nach der Stecknadel im Heuhaufen. Also müssen wir systematischer vorgehen!

Das Programm 'Speicher 4' hilft uns dabei erheblich in mehreren Stufen weiter. Sie geben ein Wort mit bis zu sechs Buchstaben ein und instruieren den Computer, wo er nach diesem Wort im Speicher suchen soll. Soll er im ganzen Speicher suchen, dauert es - bald - eine Ewigkeit. Soll er in einem begrenzten von Ihnen zu bestimmenden Speicherplatzrahmen suchen, dauert es dementsprechend lang oder kurz. Als Ergebnis bekommen Sie in jedem dieser beiden Fälle, vorausgesetzt der Suchbegriff läßt sich auffinden, die Speicherplatzstelle genannt. Spaßeshalber können

Sie dann durch 'POKE' die angegebene Stelle, (kleiner als 127, größer als 31) in Ihrem Programm direkt über die entsprechende Speicherstelle ändern. Ein Beispiel:

In Zeile 80 steht das Wörtchen 'Laenge'. Zuerst suchen wir uns die entsprechende Speicherstelle heraus. Wie? Programm starten mit 'RUN', Zeichen eingeben = 'Laenge' (ENTER), im ganzen Speicher suchen lassen = 'N' auf die naechste Frage hin eintippen (ENTER). Nach kurzer Zeit erscheint auf dem Bildschirm: 'Laenge an Speicherstelle 682 gefunden'. Kommt bei Ihnen eine etwas andere Zahl heraus, so ist das gar nicht schlimm. Wir nennen die gefundene Zahl 'ZAHL' (in unserem Fall 'Zahl' = 682). Schauen wir mal nach, was bei 'ZAHL' im Speicher steht: 'PRINT PEEK(ZAHL)' (Bitte nicht 'ZAHL' eintippen, sondern die Zahl an dieser Stelle verwenden, die Ihr CPC vorhin ausgegeben hat - bei mir also: 'PRINT PEEK(682)'). Ergebnis? '76'! Und was bedeutet in unserer Sprache die Zahl 76? 'PRINT CHR$(76)' = 'L'!

Also die Zahl 76 steht für den ersten Buchstaben unseres vorhin gesuchten Wortes 'Laenge'. Listen wir Zeile 80, so steht dort 'Laenge'. Mit 'POKE ZAHL,andere Zahl als 76' werden wir nun das 'L' von Laenge in unserem BASIC-Programm z.B. in ein 'G' verwandeln. Nur welche Zahl steht beim Computer für den Buchstaben 'G'? Ganz einfach rauszukriegen über den Befehl 'ASC('. Also: 'PRINT ASC("G")' = 71. Nun einfach eingeben 'POKE ZAHL,71' und anschließend noch einmal Zeile 80 listen -) 'Laenge' wurde zu 'Gaenge'!

So kann man ein klein wenig mehr über die Abspeicherung von BASIC-Programmen dazulernen und vielleicht dadurch noch eingehender die Komplexität eines Computers wie des CPC erahnen lernen (BASIC ist wohlgemerkt erst das Endprodukt zahlreicher Verknüpfungen).

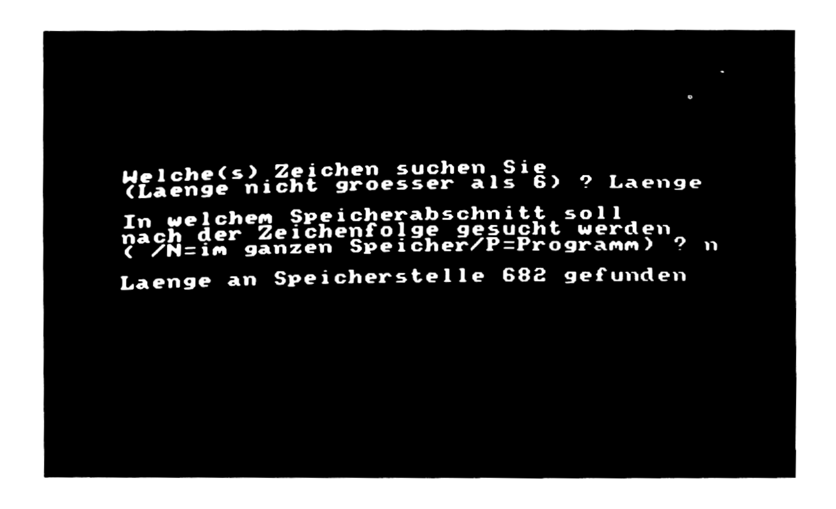
Aber noch eine weitere Dimension wird mit dem Programm 'Speicher 4' erschlossen: Die Abspeicherung der Zeilennummer. Starten Sie hierfür noch einmal das Programm mit 'RUN', geben wieder den Suchbegriff 'Laenge' (bzw. nun 'Gaenge') ein und lassen im 'P=Programm' danach suchen. Nach kurzer Zeit steht nicht nur die

entsprechende Speicherstelle wieder auf dem Bildschirm, nein auch die entsprechende Zeilennummer in unserem BASIC-Programm.

Dieses Errechnen geht verhältnismäßig einfach vor sich ... wenn man weiß wie. Jedes CPC-BASIC-Programm beginnt im Normalfall bei der Speicherstelle 368. In Speicherstelle 368 und 369 steht die Längenangabe der ersten Zeilennummer unseres Programms ('PRINT PEEK(368)' = 35; 'PRINT PEEK(369)' = 0). Die Errechnung geht folgendermaßen vor sich: 'PRINT PEEK(368) + (PEEK(369)*256)' = 35. Also: die erste Proramnzeile (inkl. Seicherstellen 368 und 369) ist 35 Bytes lang (daraus kann man schon den Beginn der nächsten Programmzeile, die auch über entsprechende Angaben verfügt, errechnen). In Speicherstelle 370 und 371 ist die Zeilennummer der ersten Programmzeile gespeichert: 'PRINT PEEK(370)' = 10; 'PRINT PEEK(371)' = 0. Errechnung s.o.: 'PRINT PEEK(370) + (PEEK(371)*256)' = 10. Also: die erste Zeile unseres Programms 'Speicher 4' ist Zeile Nummer 10 und sie ist 35 Bytes lang.

Genau diese Berechnungen führt der CPC immer und immer wieder (ab Zeile 160) aus, um uns beim Auffinden des Suchbegriffs auch die Zeilennummer anzeigen zu können.

Nun aber genug der grauen Theorie! Gehen wir zu 'Speicher 5' über.



```
10 REM Untersuchung des Speichers 4
20 REM Zeichenfolge im Speicher suchen
30 REM CPC464 Basic Programme
40 REM Copyright 1984 DATA BECKER &
                         Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
60 MODE 1
70 REM Eingabe des Suchbegriffs
80 INPUT "Welche(s) Zeichen suchen Sie
          (Laenge nicht groesser als 6)
";f$
90 IF LEN(f$)>6 OR LEN(f$)=0 THEN GOSUB
540:GOTO 60
100 PRINT
110 REM Speicherraum fuer Suche waehlen
120 INPUT "In welchem Speicherabschnitt
soll       nach der Zeichenfolge gesucht
 werden     ( /N=im ganzen Speicher/P=Pro
gramm) ";f1$
130 f1$=UPPER$(f1$)
140 PRINT
150 IF f1$="N" THEN a=0:e=65535:GOTO 370
160 IF f1$="P" THEN a=368:GOTO 180 ELSE
GOTO 300
170 REM Suche nach eingegebenem Such-
        begriff nur im Programmrahmen;
        wenn gefunden -> Anzeige der
        entsprechenden Zeilennummer
180 aa=PEEK(a)+(PEEK(a+1)*256):bb=PEEK(a
+2)+(PEEK(a+3)*256)
190 FOR n=a TO a+aa
200 IF CHR$(PEEK(n))=MID$(f$,1,1)THEN k=
1
210 IF k=1 AND CHR$(PEEK(n+1))=MID$(f$,2
,1)THEN k=2
220 IF k=2 AND CHR$(PEEK(n+2))=MID$(f$,3
,1)THEN k=3
230 IF k=3 AND CHR$(PEEK(n+3))=MID$(f$,4
,1)THEN k=4
```

```
240 IF k=4 AND CHR$(PEEK(n+4))=MID$(f$,5
,1)THEN k=5
250 IF k=5 AND CHR$(PEEK(n+5))=MID$(f$,6
,1)THEN k=6
260 IF k=LEN(f$) THEN PRINT f$;" in Zeil
e";bb;"gefunden (";n;")"
270 k=0
280 NEXT n
290 a=a+aa:IF aa=0 OR bb=0 THEN GOTO 470
 ELSE GOTO 180
300 PRINT
310 REM Darstellungsraum eingrenzen
320 INPUT "Anfangsadresse ";a
330 INPUT "Endadresse ";e
340 IF a>e OR a<0 OR e<0 OR a>65535 OR e
>65535 THEN GOSUB 540:GOTO 320
350 PRINT
360 REM Suche im Speicher von Adresse
        a bis e nach dem Suchbegriff f$
370 FOR z=a TO e
380 IF CHR$(PEEK(z))=MID$(f$,1,1)THEN k=
1
390 IF k=1 AND CHR$(PEEK(z+1))=MID$(f$,2
,1)THEN k=2
400 IF k=2 AND CHR$(PEEK(z+2))=MID$(f$,3
,1)THEN k=3
410 IF k=3 AND CHR$(PEEK(z+3))=MID$(f$,4
,1)THEN k=4
420 IF k=4 AND CHR$(PEEK(z+4))=MID$(f$,5
,1)THEN k=5
430 IF k=5 AND CHR$(PEEK(z+5))=MID$(f$,6
,1)THEN k=6
440 IF k=LEN(f$) THEN PRINT f$;" an Spei
cherstelle";z;"gefunden"
450 k=0
460 NEXT z
470 PRINT
480 PRINT
```

```
490 REM Programmende oder Fortfahren
500 INPUT "Suchen Sie weitere Zeichen (
/N) ";f$
510 f$=UPPER$(f$)
520 IF f$<>"N" THEN GOTO 60
530 END
540 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
550 PEN normal:GOSUB 560:RETURN
560 PRINT:PRINT TAB(7) "<Bitte eine Tast
e druecken>"
570 f$=INKEY$:IF f$="" THEN GOTO 570
580 PRINT:RETURN
```

Speicher 5

Während wir im Programm 'Speicher 3' jeweils in einer Zeile acht Bytes dargestellt haben (hexadezimal und als CHR$-Wert), wird in diesem Programm alles übersichtlicher = ein Byte pro Zeile und zugleich vielseitiger dargestellt.

Die Darstellungsformen dezimal und hexadezimal laufen jeweils nebeneinander:

1) Adresse dezimal
2) Adresse hexadezimal
3) Inhalt der Adresse dezimal
4) Inhalt der Adresse hexadezimal
5) Inhalt der Adresse (wenn darstellbar) als 'CHR$('

Zu Ihrer eigenen Speicherdurchforstung ist dieses Programm recht sinnvoll, denn Sie brauchen nicht immer alles umzurechnen; außerdem bietet dieses Programm eine ideale Möglichkeit, die Vielzahl der Tokens (siehe Anhang) kennenzulernen (erinnern Sie sich noch? Token = BASIC-Wort).

Machen wir einen Versuch: Ergänzen Sie das Programm um Zeile 9: '9 PRINT' (ENTER). Wie Sie wissen, beginnt der BASIC-Programmspeicher bei Adresse 368. Geben wir also als Anfangsadresse '368' ein; der Endwert soll uns egal sein, soll aber in jedem Fall größer als die Anfangsadresse sein (druecken wir nur (ENTER), so läuft die Speicherabbildung bis Anfangsadresse+2000 = 2368). Es reicht für unser Experiment, wenn wir uns nur die ersten 10 Speicherplätze anzeigen lassen, also Endadresse 378! Listen Sie bitte zusätzlich zur Kontrolle Zeile 9 und Zeile 10.

Kurz zur Deutung des Speicherabbildes:

Adresse 368: Inhalt 6 (368 und 369 geben die Zeilenlaenge an)
Adresse 369: Inhalt 0 (Adr.368+(256*Adr.369) = 6 Bytes)
Adresse 370: Inhalt 9 (370 und 371 geben die Zeilennr. an)
Adresse 371: Inhalt 0 (Adr.370+(256*Adr.371) = Zeilennr. 9)
Adresse 372: Inhalt 191 (Tokenwert für den Befehl 'PRINT')

Adresse 373: Inhalt 0   (Zeilenende-Anzeige)
Adresse 374: Inhalt 35 (374 und 375 geben die Zeilenlaenge an)
Adresse 375: Inhalt 0   (Adr.374+(256*Adr.375) = 35 Bytes)
Adresse 376: Inhalt 10 (376 und 377 geben die Zeilennr. an)
Adresse 377: Inhalt 0   (Adr.376+(256*Adr.377) = Zeilennr. 10)
Adresse 378: Inhalt 197 (Tokenwert für den Befehl 'REM')

Interessieren soll uns in diesem Zusammenhang nur Adresse 372. Geben Sie bitte einmal 'POKE 372,197' ('197' = Tokenzahl für 'REM') ein und listen anschließend Zeile 9. Nun steht dort nicht mehr wie ursprünglich 'PRINT', dafür aber viel besser 'REM'. Es gibt nicht nur Einbyte-('PRINT' = 191, 'REM' = 197)Tokens, es gibt auch Zweibytetokens. Aber davon hier nicht mehr. Schauen Sie zur Orientierung bitte im Anhang nach!



```
10 REM Untersuchung des Speichers 5
20 REM Speicherausgabe Byte fuer Byte
       pro Zeile
30 REM CPC464 Basic Programme
40 REM Copyright 1984 DATA BECKER &
                            Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
60 MODE 1
70 REM Darstellungsrahmen eingrenzen
80 INPUT "Anfangsadresse ";a
90 REM
100 PRINT "Endadresse (Zahl/<ENTER>->";a
+2000;")":INPUT e
110 IF a<0 OR e<0 OR a>65535 OR e>65535
THEN GOSUB 270:GOTO 60
120 IF e=0 THEN e=a+2000
130 CLS
140 REM Reihenfolge im Ausdruck:
        Adresse dezimal
        Adresse hexadezimal
        Inhalt der Adresse dezimal
        Inhalt der Adresse hexadezimal
        Inhalt der Adresse als CHR$(
150 FOR z=a TO e
160 PRINT z;
170 PRINT TAB(8) HEX$(z,4);
180 PRINT TAB (18) PEEK(z);
190 PRINT TAB(24) HEX$(PEEK(z),2);
200 PRINT TAB(30);
210 n1=PEEK(z)
220 IF n1<32 THEN n1=46
230 IF n1>126 THEN n1=46
240 PRINT CHR$(n1)
250 NEXT z
260 END
```

```
270 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
280 PEN normal:GOSUB 290:RETURN
290 PRINT:PRINT TAB(7) "<Bitte eine Tast
e druecken>"
300 f$=INKEY$:IF f$="" THEN GOTO 300
310 RETURN
```

Grafikeditor

Nicht nur der Sound des CPC läßt sich ohne Handbuch nur mit Schwierigkeiten programmieren, ähnlich wenn nicht sogar gleich verhält es sich mit der Grafikprogrammierung beim CPC.

Einerseits gibt es lediglich zwei direkte Grafikbefehle (Punkt setzen = 'PLOT', Linie zeichnen = 'DRAW'), andererseits muß man sich schon gut auf einem Koordinatenkreuz zu Hause fühlen, will man absolut und relativ genau die gewünschten Punkte auf dem Bildschirm ansprechen.

Viel einfacher, schneller und auch komfortabler geht es da doch mit unserem 'Grafikeditor'.

Lange haben wir überlegt, welche Ansteuerung für den Cursor ideal wäre (nur die absolute Adressierung zu verwenden wäre in unseren Augen sinnlos, dann brauchten wir keinen Grafikeditor!). Der Joystick mußte ebenalls wie die Cursorsteuerung schon bald abgeschrieben werden, da hierdurch nur jeweils vier Himmelsrichtungen angesprochen bzw. unterschieden werden können.

Da für uns ein Achter-Richtungsblock ideal war (alle Himmelsrichtungen plus Diagonalen), wurde einfach ein Großteil der Zehnertastatur als Cursor mit acht Richtungen umfunktioniert bzw. umbelegt.

Das Programm 'Grafikeditor' bietet nicht nur die Möglichkeit des einfachen Zeichnens in acht Himmelsrichtungen:

mit 'C' können Sie den Cursor absolut setzen (z.B.'320,200' = Mittelpunkt)

mit 'D' kann eine Bildschirmfläche dupliziert werden (Sie bestimmen mit dem ersten Cursor die linke untere Ecke, mit dem zweiten Cursor die rechte obere Ecke der zu kopierenden Fläche; schließlich wird mit dem dritten Cursor die linke untere Ecke angepeilt, wo unser Duplikat auf dem Bildschirm erscheinen soll). Mit der Zusatzfunktion 'OR' wird im Gegensatz zu 'AND'

der Bilduntergrund nicht gelöscht. Schließlich können Sie das Original auch noch zweifach vergrößert darstellen
mit 'F' können Sie aus dem zur Verfügung stehenden Farbspektrum auswählen

mit 'G' läßt sich leicht eine Gerade beliebiger Länge zwischen zwei Punkten ziehen

mit 'K' wird nach Eingabe des Radius um den Cursor, der nun Mittelpunkt ist, ein Kreis gezogen

mit 'L' loeschen Sie eine durch zwei Cursoreingaben (links unten und rechts oben) bestimmte Fläche

mit 'R' (wieder links unten und rechts oben bestimmen) wird ein Rechteck auf den Bildschirm gezeichnet

mit 'T' wird schließlich ein Text an der Grafikcursorposition ausgegeben.

Sie sehen, bis auf das Ausmalen von Flächen, eine Abspeicherng des entstandenen Bildes auf Kassette (16 K dauern mit Kassette einfach zu lange, warten wir lieber auf die Diskettenstation) und eine Ausgaberoutine an den Drucker (Hardcopy) ist fast alles grafisch mit diesem 'Grafikeditor' möglich.

Vielleicht fallen Ihnen noch andere Dinge ein. Bis auf den begrenzten Computerspeicherlatz liegt Ihrer Kreativität somit kein Stein mehr im Wege.

```
10 REM Grafikeditor
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                              Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
50 MODE 1
60 FOR n=3 TO 20:KEY DEF n,1:NEXT:SPEED
KEY 50,2
70 PAPER 0:PEN normal
80 INPUT "Welchen Bildschirmmode ";f$
90 IF VAL(f$)<0 OR VAL(f$)>2 THEN GOSUB
1570:GOTO 80
100 IF VAL(f$)=0 THEN z1=4
110 IF VAL(f$)=1 THEN z1=2
120 IF VAL(f$)=2 THEN z1=1
130 MODE VAL(f$)
140 REM Festlegen des Grafik- und des
        Textfensters
150 ORIGIN 0,0,0,640,400,20
160 WINDOW #1,1,40,25,25
170 a=320:b=200
180 PAPER #1,3:CLS #1
190 PAPER 0:CLG:PLOT a,b,1
200 REM Menuevorgabe
        Auswahlmoeglichkeiten: CDFGKLRT
        C = Cursor mit Koordinaten best.
        D = Duplizieren (Or/And/Vergr.)
        F = Farbe bestimmen
        G = Gerade zeichnen
210 REM K = Kreis zeichnen
        L = Loeschen
        R = Rechteck zeichnen
        T = Text eingeben
220 CLS #1:PRINT #1,a;b;"CDFGKLRT";
230 a$=INKEY$
240 IF a$="" THEN GOTO 230
250 GOSUB 1570
260 DRAW a,b
270 IF ASC(a$)>57 THEN GOSUB 290
```

```
280 GOTO 220
290 CLS #1:PRINT #1,a;b;
300 a$=UPPER$(a$)
310 IF a$="C" THEN PRINT #1,"Curs.best."
;:zz=2
320 IF a$="D" THEN PRINT #1,"Duplizier."
;:zz=3
330 IF a$="F" THEN PRINT #1,"Farb.best."
;:zz=4
340 IF a$="G" THEN PRINT #1,"Gerade ze."
;:zz=5
350 IF a$="K" THEN PRINT #1,"Kreis zei."
;:zz=6
360 IF a$="L" THEN PRINT #1,"Loeschen  "
;:zz=7
370 IF a$="R" THEN PRINT #1,"Rechteck  "
;:zz=8
380 IF a$="T" THEN PRINT #1,"Text eing."
;:zz=9
390 IF zz=0 THEN RETURN
400 a$=INKEY$
410 IF a$="" THEN GOTO 400
420 IF a$=CHR$(13) THEN GOTO 440
430 RETURN
440 ON zz GOSUB 410,470,520,860,900,1000
,1080,1280,1450
450 RETURN
460 REM C = Cursor mit Koordinaten best.
470 CLS #1:INPUT #1,"Wohin(x,y)";a$,b$
480 IF VAL(a$)<0 OR VAL(a$)>640 OR VAL(a
$)<20 OR VAL(a$)>400 THEN GOTO 470
490 a=VAL(a$):b=VAL(b$):PLOT a,b
500 RETURN
510 REM D = Duplizieren (Or/And/Vergr.)
520 punkta=a:punktb=b:CLS #1:PRINT #1,"1
.Cursor setzen"
530 a$=INKEY$
540 IF a$="" THEN GOTO 530
550 farbe=TEST(a,b):punkt1=a:punkt2=b
```

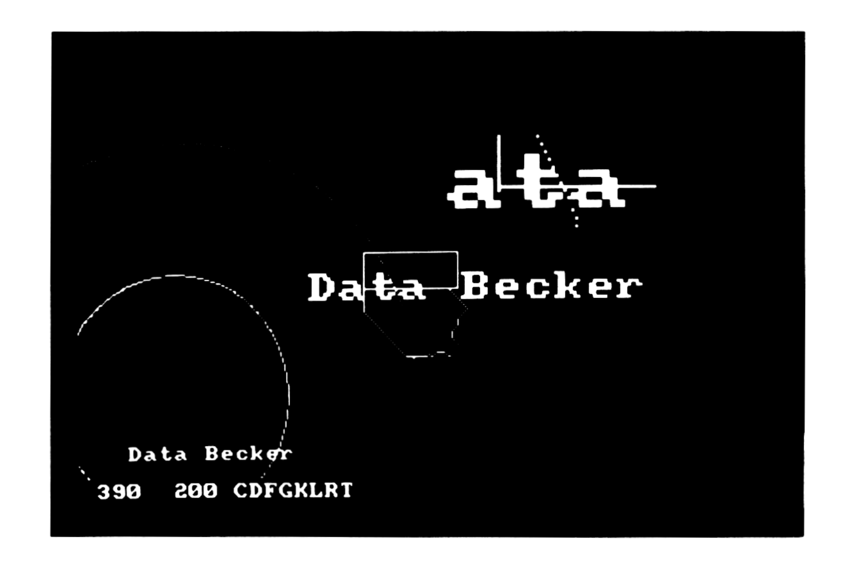
```
560 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a,
b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PLO
T a,b,farbe:NEXT m
570 GOSUB 1570
580 IF a$=CHR$(13) THEN a1=a:b1=b ELSE G
OTO 530
590 CLS #1:PRINT #1,"2.Cursor setzen"
600 a$=INKEY$
610 IF a$="" THEN GOTO 600
620 farbe=TEST(a,b):punkt1=a:punkt2=b
630 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a,
b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PLO
T a,b,farbe:NEXT m
640 GOSUB 1570
650 IF a$=CHR$(13) THEN a2=a:b2=b:IF a2<
=a1 OR b2<=b1 THEN GOTO 590 ELSE GOTO 66
0 ELSE GOTO 600
660 CLS #1:PRINT #1,"3.Cursor setzen"
670 a$=INKEY$
680 IF a$="" THEN GOTO 670
690 farbe=TEST(a,b):punkt1=a:punkt2=b
700 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a,
b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PLO
T a,b,farbe:NEXT m
710 GOSUB 1570
720 IF a$=CHR$(13) THEN a3=a:b3=b ELSE G
OTO 670
730 CLS #1:INPUT #1,"O(r) oder N(ot) ";f
$:f$=UPPER$(f$):IF LEFT$(f$,1)="O" THEN
flag=1 ELSE flag=0
740 CLS #1:INPUT #1,"vergroessert (J/ )
";f$:f$=UPPER$(f$):IF LEFT$(f$,1)="J" TH
EN flag2=1 ELSE flag2=0
750 IF flag2<>0 THEN GOTO 800
760 n1=0:m1=0:FOR n=a1 TO a2 STEP z1:FOR
 m=b1 TO b2 STEP 2
770 farbtest=TEST(n,m):IF flag=0 THEN PL
OT a3+n1,b3+m1,farbtest ELSE IF TEST(a3+
n1,b3+m1)=0 THEN PLOT a3+n1,b3+m1,farbte
st
```

```
780 m1=m1+2:NEXT m:m1=Ø:n1=n1+z1:NEXT n
790 a=punkta:b=punktb:PLOT a,b,1:RETURN
800 n1=Ø:m1=Ø:FOR n=a1 TO a2 STEP z1:FOR
 m=b1 TO b2 STEP 2
810 ft=TEST(n,m):IF flag=Ø THEN PLOT a3+
n1,b3+m1,ft:PLOT a3+n1,b3+m1+2,ft:PLOT a
3+n1+z1,b3+m1,ft:PLOT a3+n1+z1,b3+m1+2,f
t
820 IF flag=1 AND TEST(a3+n1,b3+m1)=Ø TH
EN PLOT a3+n1,b3+m1,ft:PLOT a3+n1,b3+m1+
2,ft:PLOT a3+n1+z1,b3+m1,ft:PLOT a3+n1+z
1,b3+m1+2,ft
830 m1=m1+4:NEXT m:m1=Ø:n1=n1+2*z1:NEXT
n
840 a=punkta:b=punktb:PLOT a,b,1:RETURN
850 REM F = Farbe bestimmen
860 CLS #1:INPUT #1,"Welche Nummer ";f$
870 IF VAL(f$)<Ø OR VAL(f$)>z1↑2 THEN GO
TO 860 ELSE PLOT a,b,VAL(f$)
880 RETURN
890 REM G = Gerade zeichnen
900 CLS #1:PRINT #1,"Bitte Cursor setzen
":punkt1=a:punkt2=b
910 a$=INKEY$
920 IF a$="" THEN GOTO 910
930 farbe=TEST(a,b)
940 FOR m=1 TO 1Ø:PLOT a,b,farbe:PLOT a,
b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PLO
T a,b,farbe:NEXT m
950 GOSUB 157Ø
960 IF a$=CHR$(13) THEN GOTO 98Ø
970 GOTO 91Ø
980 PLOT punkt1,punkt2,1:DRAW a,b,1:RETU
RN
990 REM K = Kreis zeichnen
1000 CLS #1:INPUT #1,"Radius ";f$
1010 FOR aa=1 TO 36Ø
1020 DEG
1030 PLOT a+VAL(f$)*COS(aa),b+VAL(f$)*SI
N(aa),1
```

```
1040 NEXT aa
1050 PLOT a,b
1060 RETURN
1070 REM L = Loeschen
1080 CLS #1:PRINT #1,"1.Cursor setzen"
1090 merker1=a:merker2=b
1100 a$=INKEY$
1110 IF a$="" THEN GOTO 1100
1120 farbe=TEST(a,b):punkt1=a:punkt2=b
1130 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a
,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PL
OT a,b,farbe:NEXT m
1140 GOSUB 1570
1150 IF a$=CHR$(13) THEN a1=a:b1=b ELSE
GOTO 1100
1160 CLS #1:PRINT #1,"2.Cursor setzen"
1170 a$=INKEY$
1180 IF a$="" THEN GOTO 1170
1190 farbe=TEST(a,b):punkt1=a:punkt2=b
1200 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a
,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PL
OT a,b,farbe:NEXT m
1210 GOSUB 1570
1220 IF a$=CHR$(13) THEN a2=a:b2=b ELSE
GOTO 1170
1230 ORIGIN 0,0,a1,a2,b2,b1:CLG:
1240 ORIGIN 0,0,0,640,400,20
1250 a=merker1:b=merker2:PLOT a,b,1
1260 RETURN
1270 REM R = Rechteck zeichnen
1280 CLS #1:PRINT #1,"1.Cursor setzen"
1290 merker1=a:merker2=b
1300 a$=INKEY$
1310 IF a$="" THEN GOTO 1300
1320 farbe=TEST(a,b):punkt1=a:punkt2=b
1330 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a
,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PL
OT a,b,farbe:NEXT m
1340 GOSUB 1570
```

```
1350 IF a$=CHR$(13) THEN a1=a:b1=b ELSE
GOTO 1300
1360 CLS #1:PRINT #1,"2.Cursor setzen"
1370 a$=INKEY$
1380 IF a$="" THEN GOTO 1370
1390 farbe=TEST(a,b):punkt1=a:punkt2=b
1400 FOR m=1 TO 10:PLOT a,b,farbe:PLOT a
,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:PL
OT a,b,farbe:NEXT m
1410 GOSUB 1570
1420 IF a$=CHR$(13) THEN a2=a:b2=b ELSE
GOTO 1370
1430 PLOT a1,b1:DRAW a2,b1,1:DRAW a2,b2,
1:DRAW a1,b2,1:DRAW a1,b1,1:a=merker1:b=
merker2:PLOT a,b,1:RETURN
1440 REM T = Text eingeben
1450 CLS #1:PRINT #1,"Cursor setzen":mer
ker1=a:merker2=b
1460 a$=INKEY$
1470 IF a$="" THEN GOTO 1460
1480 farbe=TEST(a,b):punkt1=a:punkt2=b:F
OR m=1 TO 10:PLOT a,b,farbe:PLOT a,b,1:P
LOT a,b,2:PLOT a,b,3:PLOT a,b,1:PLOT a,b
,farbe:NEXT m
1490 GOSUB 1570
1500 IF a$=CHR$(13) THEN a1=a:b1=b:PLOT
a1,b1,1 ELSE GOTO 1460
1510 TAG
1520 CLS #1:INPUT #1,"Text ";a$
1530 PRINT a$;
1540 a=merker1:b=merker2:PLOT a,b,1
1550 TAGOFF:RETURN
1560 REM Ausrechnen der Cursorbe-
         wegung (8 Richtungen)
1570 IF ASC(a$)=54 THEN a=a+z1
1580 IF ASC(a$)=57 THEN a=a+z1:b=b+z1
1590 IF ASC(a$)=56 THEN b=b+2
1600 IF ASC(a$)=55 THEN a=a-z1:b=b+z1
1610 IF ASC(a$)=52 THEN a=a-z1
1620 IF ASC(a$)=49 THEN a=a-z1:b=b-z1
```

```
1630 IF ASC(a$)=50 THEN b=b-2
1640 IF ASC(a$)=51 THEN a=a+z1:b=b-z1
1650 RETURN
1660 a$=INKEY$
1670 IF a$="" THEN 1660
1680 PRINT ASC(a$)
1690 GOTO 1660
```



Soundeditor

Der CPC hat zwar einen tollen Soundprozessor, jedoch wird der Umgang mit dem vielen unverständlichen Zahlwerk schon so manchem begeisterten Freak nach langem Versuchen den Wind aus den Segeln genommen haben. Wie kann man sich auch logisch erklären, daß die Tonkanäle beim SOUND-Befehl an erster Stelle angesprochen werden, die Lautstärke jedoch erst an vierter Stelle? Wie merkt man sich die vielfältigen Rendezvous-Techniken?

Unserer Meinung nach ist da nicht mit Auswendiglernen weitergeholfen, vielmehr läuft es hierbei immer mehr auf das häufige Nachschlagen im Handbuch hinaus!

Oder: Wie merken wir uns, welche der phantastischen Ton- bzw. Lautstärkekurven gerade aktiv ist nd welche wir aktivieren können? Notitzen auf Papier machen?

Wir haben mit unserem 'Soundeditor' eine komfortablere Lösung anzubieten. Sie finden sämtliche Parameter mit ihrer deutschen Bedeutung auf einer Bildschirmseite und können nun durch Drücken der Taste (SHIFT) mit einem Buchstaben zusammen anheben (z.B. (SHIFT) 'H' -) Tonhöhe steigern) oder durch Drücken der Taste (CTRL) mit einem Buchstaben zusammen senken (z.B. (CTRL) 'L' -) Latstärke verringern).

So läßt sich so manch ein Ton oder manches Geräusch in aller Ruhe am Bildschirm editieren und durch anschließendes Drücken der Taste (ENTER) auch aktivieren (denken Sie bitte daran, daß für die Erzeugung eines Tones bei diesem Soundeditor die Lautstärke, die Tonhöhe und zumindest ein Tonkanal aktiviert sein muß).

Bis zu sechzehn verschiedene Hüllkurven jeweils getrennt für Ton und Lautstärke können zusätzlich aktiviert und bei der Toneditierung eingesetzt werden. Nach dem Drücken der Tasten (SHIFT) und 'N' (ENV-Eingabe) werden Sie nach der ENV-Nummer und der Parameterzahl befragt. Die ENV-Nummer hat etwas mit der Programmabspeicherung zu tun und ist nicht unbedingt gleichbedeu-

tend mit der Folgenummer der Hüllkurve. Somit beträgt die Mindestparameteranzahl, nach der im Programm gefragt wird, auch nicht drei sondern vier.

Ähnlich werden ENT-Hüllkurven erzeugt (Tonhüllkurven) und zwar durch gleichzeitiges Niederdrücken der Tasten (SHIFT) und 'O'. Schließlich können wir uns einen Überblick über die erstellten Hüllkurven durch (SHIFT) und 'P' verschaffen.

Viel Spaß bei der Editierung neuer Klänge mit Ihrem CPC und diesem Soundeditor!

```
Der          beim Schneider CPC464
   <SHIFT> -> +     <CTRL> -> -
          A:Kanal A
          B:Kanal B
          C:Kanal C aus
D:Rendezvous mit Kanal A
E:Rendezvous mit Kanal B aus
F:Rendezvous mit Kanal C aus
    G:Geraeusch Periode aus
          H:Tonhoehe
I:Tondauer in 1/100 Sek.
J:Abruf einer ENV-Huellkurve
K:Abruf einer ENT-Huellkurve 0
          L:Lautstaerke
N:ENV-Eing.  O:ENT-Eing.  P:Liste
```

```
10 REM Soundeditor
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24
50 DIM a(15,15),a1(15,15)
60 MODE 1
70 SPEED INK 20,20
80 PEN 1
90 REM Bildschirmaufbau mit Angaben zu
        Kanal, Huellkurve, Tonhoehe ...
100 PRINT"   Der ";:PEN 2:PRINT"Sound";:
PEN 1:PRINT" beim Schneider CPC464"
110 PRINT
120 PRINT"      <SHIFT> -> +   <CTRL> ->
_"
130 PRINT
140 PRINT TAB(11) "A:Kanal A ";:IF a=0 T
HEN PRINT"aus" ELSE PEN 2:PRINT"an ":PEN
 1
150 PRINT TAB(11) "B:Kanal B ";:IF b=0 T
HEN PRINT"aus" ELSE PEN 2:PRINT"an ":PEN
 1
160 PRINT TAB(11) "C:Kanal C ";:IF c=0 T
HEN PRINT"aus" ELSE PEN 2:PRINT"an ":PEN
 1
170 PRINT
180 PRINT TAB(4) "D:Rendezvous mit Kanal
 A ";:IF d=0 THEN PRINT"aus" ELSE PEN 2:
PRINT"an":PEN 1
190 PRINT TAB(4) "E:Rendezvous mit Kanal
 B ";:IF e=0 THEN PRINT"aus" ELSE PEN 2:
PRINT"an":PEN 1
200 PRINT TAB(4) "F:Rendezvous mit Kanal
 C ";:IF f=0 THEN PRINT"aus" ELSE PEN 2:
PRINT"an":PEN 1
210 PRINT
220 PRINT TAB(7) "G:Geraeusch Periode ";
:IF g=0 THEN LOCATE 27,13:PRINT "aus" EL
SE PEN 2:LOCATE 27,13:PRINT g:PEN 1
```

```
230 PRINT
240 PRINT TAB(11) "H:Tonhoehe ";:IF h=0
THEN LOCATE 22,15:PRINT"aus" ELSE PEN 2:
LOCATE 22,15:PRINT h:PEN 1
250 PRINT
260 PRINT TAB(4) "I:Tondauer in 1/100 Se
k.";:IF i=0 THEN PRINT"aus" ELSE PEN 2:L
OCATE 28,17:PRINT i:PEN 1
270 PRINT
280 PRINT TAB(4) "J:Abruf einer ENV-Huel
lkurve ";:IF j=0 THEN LOCATE 33,19:PRINT
"0" ELSE PEN 2:LOCATE 32,19:PRINT j:PEN
1
290 PRINT TAB(4) "K:Abruf einer ENT-Huel
lkurve ";:IF k=0 THEN LOCATE 33,20:PRINT
"0" ELSE PEN 2:LOCATE 32,20:PRINT k:PEN
1
300 PRINT
310 PRINT TAB(11) "L:Lautstaerke ";:IF l
=0 THEN LOCATE 25,22:PRINT"aus" ELSE PEN
 2:LOCATE 25,22:PRINT l:PEN 1
320 PRINT
330 PRINT TAB(4) "N:ENV-Eing.  O:ENT-Ein
g.  P:Liste"
340 a$=INKEY$
350 REM Soundkanal und Rendevous wie
        eingestellt uebernehmen
360 a1=a+b+c+d+e+f
370 REM Tastaturabfrage inklusive
        Tonerzeugung, wenn <ENTER>
        gedrueckt
380 IF a$=CHR$(13) THEN SOUND a1,h,i,l,j
,k,g
390 IF a$=CHR$(14) OR a$=CHR$(78) THEN G
OTO 2120
400 IF a$=CHR$(15) OR a$=CHR$(79) THEN G
OTO 2250
410 IF a$=CHR$(16) OR a$=CHR$(80) THEN G
OTO 2370
420 IF a$="" THEN GOTO 340
```

```
430 IF ASC(a$)>12 AND ASC(a$)<65 THEN GO
TO 340
440 IF ASC(a$)>76 THEN GOTO 340
450 IF ASC(a$)>12 THEN GOTO 1250
460 REM Falls die gedrueckte Taste
        eine Aenderung auf dem Bild-
        schirm bewirkt hat, wird sie
        durch folgende Zeilen erzeugt
470 ON ASC(a$) GOSUB 490,530,570,610,650
,690,730,820,910,1000,1080,1160
480 GOTO 340
490 a=0
500 LOCATE 21,5
510 PRINT"aus"
520 RETURN
530 b=0
540 LOCATE 21,6
550 PRINT"aus"
560 RETURN
570 c=0
580 LOCATE 21,7
590 PRINT"aus"
600 RETURN
610 d=0
620 LOCATE 29,9
630 PRINT"aus"
640 RETURN
650 e=0
660 LOCATE 29,10
670 PRINT"aus"
680 RETURN
690 f=0
700 LOCATE 29,11
710 PRINT"aus"
720 RETURN
730 IF g>0 THEN g=g-1
740 IF g=0 THEN LOCATE 27,13:PRINT"aus":
RETURN
750 LOCATE 27,13
760 PRINT"   "
```

```
770 LOCATE 27,13
780 PEN 2
790 PRINT g
800 PEN 1
810 RETURN
820 IF h>0 THEN h=h-1
830 IF h=0 THEN LOCATE 22,15:PRINT"aus":
RETURN
840 LOCATE 22,15
850 PRINT"    "
860 LOCATE 22,15
870 PEN 2
880 PRINT h
890 PEN 1
900 RETURN
910 IF i>0 THEN i=i-1
920 IF i=0 THEN LOCATE 28,17:PRINT"aus":
RETURN
930 LOCATE 28,17
940 PRINT"     "
950 LOCATE 28,17
960 PEN 2
970 PRINT i
980 PEN 1
990 RETURN
1000 IF j>0 THEN j=j-1
1010 LOCATE 32,19
1020 PRINT"  "
1030 LOCATE 32,19
1040 IF j>0 THEN PEN 2
1050 PRINT j
1060 PEN 1
1070 RETURN
1080 IF k>0 THEN k=k-1
1090 LOCATE 32,20
1100 PRINT"  "
1110 LOCATE 32,20
1120 IF k>0 THEN PEN 2
1130 PRINT k
1140 PEN 1
```

```
1150 RETURN
1160 IF l>0 THEN l=l-1
1170 LOCATE 25,22
1180 PRINT"    "
1190 LOCATE 25,22
1200 IF l=0 THEN PRINT"aus":RETURN
1210 PEN 2
1220 PRINT l
1230 PEN 1
1240 RETURN
1250 ON (ASC(a$)-64) GOSUB 1270,1330,139
0,1450,1510,1570,1630,1710,1790,1870,195
0,2030
1260 GOTO 340
1270 a=1
1280 LOCATE 21,5
1290 PEN 2
1300 PRINT"an "
1310 PEN 1
1320 RETURN
1330 b=2
1340 LOCATE 21,6
1350 PEN 2
1360 PRINT"an "
1370 PEN 1
1380 RETURN
1390 c=4
1400 LOCATE 21,7
1410 PEN 2
1420 PRINT"an "
1430 PEN 1
1440 RETURN
1450 d=8
1460 LOCATE 29,9
1470 PEN 2
1480 PRINT"an "
1490 PEN 1
1500 RETURN
1510 e=16
1520 LOCATE 29,10
```

```
1530 PEN 2
1540 PRINT"an "
1550 PEN 1
1560 RETURN
1570 f=32
1580 LOCATE 29,11
1590 PEN 2
1600 PRINT"an "
1610 PEN 1
1620 RETURN
1630 IF g<15 THEN g=g+1
1640 LOCATE 27,13
1650 PRINT"  "
1660 LOCATE 27,13
1670 PEN 2
1680 PRINT g
1690 PEN 1
1700 RETURN
1710 IF h<4095 THEN h=h+1
1720 LOCATE 22,15
1730 PRINT"    "
1740 LOCATE 22,15
1750 PEN 2
1760 PRINT h
1770 PEN 1
1780 RETURN
1790 IF i<32767 THEN i=i+1
1800 LOCATE 28,17
1810 PRINT"     "
1820 LOCATE 28,17
1830 PEN 2
1840 PRINT i
1850 PEN 1
1860 RETURN
1870 IF j<15 THEN j=j+1
1880 LOCATE 32,19
1890 PRINT"  "
1900 LOCATE 32,19
1910 PEN 2
```

```
1920 PRINT j
1930 PEN 1
1940 RETURN
1950 IF k<15 THEN k=k+1
1960 LOCATE 32,20
1970 PRINT"  "
1980 LOCATE 32,20
1990 PEN 2
2000 PRINT k
2010 PEN 1
2020 RETURN
2030 IF l<15 THEN l=l+1
2040 LOCATE 25,22
2050 PRINT"   "
2060 LOCATE 25,22
2070 PEN 2
2080 PRINT l
2090 PEN 1
2100 RETURN
2110 REM Eine neue ENV-Huellkurve soll
          erzeugt werden
2120 LOCATE 4,24
2130 PRINT"
      "
2140 LOCATE 4,24
2150 INPUT"ENV-Nr.und Parameteranzahl ";
z,z1
2160 IF z<0 OR z>15 OR (z1<>4 AND z1<>7
AND z1<> 10 AND z1<>13 AND z1<>16) THEN
GOTO 2120 ELSE LOCATE 4,24:PRINT"
                                 ":LOCATE 4,2
4
2170 IF z1=4 THEN INPUT"4 Par. ";a(z,0),
a(z,1),a(z,2),a(z,3)
2180 IF z1=7 THEN INPUT"7 Par. ";a(z,0),
a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z,6
)
2190 IF z1=10 THEN INPUT"10 Par. ";a(z,0
),a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z
,6),a(z,7),a(z,8),a(z,9)
```

```
2200 IF z1=13 THEN INPUT"13 Par. ";a(z,Ø
),a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z
,6),a(z,7),a(z,8),a(z,9),a(z,1Ø),a(z,11)
,a(z,12)
221Ø IF z1=16 THEN INPUT"16 Par. ";a(z,Ø
),a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z
,6),a(z,7),a(z,8),a(z,9),a(z,1Ø),a(z,11)
,a(z,12),a(z,13),a(z,14),a(z,15)
222Ø ENV a(z,Ø),a(z,1),a(z,2),a(z,3),a(z
,4),a(z,5),a(z,6),a(z,7),a(z,8),a(z,9),a
(z,1Ø),a(z,11),a(z,12),a(z,13),a(z,14),a
(z,15)
223Ø GOTO 237Ø
224Ø REM Eine neue ENT-Huellkurve
         soll erzeugt werden
225Ø LOCATE 4,24
226Ø PRINT "
      "
227Ø LOCATE 4,24
228Ø INPUT"ENT-Nr.und Parameteranzahl ";
z,z1
229Ø IF z<Ø OR z>15 OR (z1<>4 AND z1<>7
AND z1<> 1Ø AND z1<>13 AND z1<>16) THEN
GOTO 225Ø ELSE LOCATE 4,24:PRINT"
                                ":LOCATE 4,2
4
23ØØ IF z1=4 THEN INPUT"4 Par. ";a1(z,Ø)
,a1(z,1),a1(z,2),a1(z,3)
231Ø IF z1=7 THEN INPUT"7 Par. ";a1(z,Ø)
,a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,5)
,a1(z,6)
232Ø IF z1=1Ø THEN INPUT"1Ø Par. ";a1(z,
Ø),a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,
5),a1(z,6),a1(z,7),a1(z,8),a1(z,9)
233Ø IF z1=13 THEN INPUT"13 Par. ";a1(z,
Ø),a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,
5),a1(z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,
1Ø),a1(z,11),a1(z,12)
```

```
2340 IF z1=16 THEN INPUT"16 Par. ";a1(z,
0),a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,
5),a1(z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,
10),a1(z,11),a1(z,12),a1(z,13),a1(z,14),
a1(z,15)
2350 ENT a1(z,0),a1(z,1),a1(z,2),a1(z,3)
,a1(z,4),a1(z,5),a1(z,6),a1(z,7),a1(z,8)
,a1(z,9),a1(z,10),a1(z,11),a1(z,12),a1(z
,13),a1(z,14),a1(z,15)
2360 REM Die vorhandenen ENV- bzw.
         ENT-Huellkurven sollen auf
         dem Bildschirm dargestellt
         werden
2370 MODE 1
2380 a$=""
2390 INPUT"ENV-Liste oder ENT-Liste (V/T
) ";a$
2400 IF a$="" THEN GOTO 60
2410 IF a$<>"V" AND a$<>"v" AND a$<>"T"
AND a$<>"t" THEN GOTO 2370
2420 IF a$<>"v" AND a$<>"V" THEN GOTO 25
30
2430 REM Ausgabe der ENV-Huellkurven
2440 FOR n=0 TO 15
2450 FOR m=0 TO 15
2460 IF a(n,0)=0 THEN GOTO 2470 ELSE PRI
NT a(n,m);
2470 NEXT m
2480 PRINT:PAPER 3:PRINT n;".";:PAPER 4
2490 NEXT n
2500 a$=INKEY$
2510 IF a$="" THEN GOTO 2500 ELSE GOTO 6
0
2520 REM Ausgabe der ENT-Huellkurven
2530 FOR n=0 TO 15
2540 FOR m=0 TO 15
2550 IF a1(n,0)=0 THEN GOTO 2560 ELSE PR
INT a1(n,m);
2560 NEXT m
2570 PRINT:PAPER 3:PRINT n;".";:PAPER 4
2580 NEXT n
2590 a$=INKEY$
2600 IF a$="" THEN GOTO 2590 ELSE GOTO 6
0
```

Texteditor
==========

Wir haben dieses Programm absichtlich nicht Textbearbeitung oder gar Textverarbeitung genannt, denn bei dieser Wahl der Bezeichnung wären die Erwartungen eines fachkundigen Lesers und Programmierers einfach zu hoch gesteckt.

Bei einer Textbearbeitung sollte man zumindest Textbausteine zusätzlich im Speicher in den bereits bestehenden Text einfügen können, und dies können wir mit einer Kassettenrecorderabspeicherung nur äußerst schwer zufriedenstellend bewältigen.

Bei einer Textverarbeitung hingegen sollte eine Kopplung mit Adreßdateien möglich sein, der deutsche Zeichensatz und die deutsche Silbentrennung sollten in das Programm mit einfließen und schließlich: die Textverarbeitung sollte unabhängig von dem zur freien Verfügung stehenden Computerspeicherpklatz zwischendurch Textteile irgendwo ablegen (normalerweise auf Diskette) und beim Zurückblättern auch diese Inhalte wieder in den Speicher automatisch laden können.

All diesem steht die bisherige Abspeicherungsmöglichkeit des CPC (mit Kassette) im Wege, geschweige denn: eine derart komfortable Textverarbeitung würde den Rahmen dieser Programmsammlung sprengen.

Nachdem wir Ihre Erwartungen heruntergeschraubt haben, nun aber doch noch die Erläuterung der Fähigkeiten unseres Texteditors: Wir können bis zu 10 Seiten a 22 Zeilen in unserem CPC abspeichern. Mit der normalen Cursorsteuerung können wir vorblättern oder zurückblättern, Texte schreiben bzw. dorthin plazieren, wo sie unserer Meinung nach stehen sollten, aber natürlich auch einzelne Buchstaben oder mehr Zeichen löschen. Sind wir mit dem Cursor irgendwo auf dem Bildschirm 'herumgelaufen', können wir ihn durch das Drücken der COPY-Taste wieder an die alte Stelle positionieren. Die Stellung des Cursors im aktuellen Text wird im Erläuterungsfenster nicht nur durch die Angabe der Koordinaten, auch durch einen Strich wird uns seine Horizontalposition

eindeutig angezeigt. Dieser 'Unterstreichungsstrich' wird jeweils im Überschreibmodus ('PRINT CHR$(22);CHR$(1)') im Erläuterungsfenster erzeugt.

Wollen wir zurückblättern, müssen wir den Cursor in Zeile 1 stehen haben und die Pfeiltaste (hoch) drücken. Wollen wir vorblättern, können wir entweder mit dem Cursor in Zeile 22 gehen und nun die entsprechende Pfeiltaste drücken (nach unten) oder wir können (CTRL) und 'B' gleichzeitig drücken, was so viel bedeutet wie 'B'lättern.

Wollen wir den eingetippten Text abspeichern, drücken wir (CTRL) und 'S' wie 'S'aven (der gewünschte Filename muß eingegeben werden), zum 'L'aden eines Textes von Kassette in den Computerspeicher (CTRL) 'L'.

Wir können im Speicher nach Inhalten suchen (auch weitersuchen lassen) und schließlich auch einen Ausdruck des Textes in die Wege leiten.

Die zuletzt genannte Funktion, das Ausdrucken von eingegebenen Texten, sei noch ein klein wenig genauer erläutert: Sie haben vorher im 40-Spalten-Format den Text eingegeben (zum Teil auch über das Zeilenende hinweg in die nächste Textzeile geschrieben). Beim Ausdrucken richtet sich der CPC nach auftretenden Leerzeichen und gibt den Text mit mindestens 40 Zeichen/Zeile (je nach Ihrer Wahl auch mehr) auf den angeschlossenen Drucker aus.

Noch ein Hinweis:

Sind Sie kein Druckerbesitzer, können Sie sich spaßeshalber den vom Computer für den Drucker angepaßten Text auch auf dem 80-Zeichen-Bildschirmmode ('MODE 2') ansehen. Hierzu müssen allerdings ein paar kleine Änderungen im Programm vorgenommen werden:

1) Ergänzung des Programms um Zeile 755:

'755 MODE 2'

2) Änderung des Befehls 'PRINT #8' in den Zeilen 780 und 790 lediglich in 'PRINT'

3) Vielleicht ist es sinnvoll, Zeile 810 vor der Umstellung auf 40 Zeichen ('MODE 1') noch durch eine Warteschleife ('FOR n=1 to 10000') oder gar durch eine Fortsetzungsaufforderung ('PRINT "Bitte eine Taste druecken!"' plus Tastenabfrage mit 'INKEY$') zu erweitern; beim Drucker hingegen ist eine sofortige Rückkehr ins Programm gut so, denn zu diesem Zeitpunkt haben wir ja unsere Texte oder Daten bereits schwarz auf weiß!

Nun können wir herumexperimentieren, wie der Text z.B. in 80, 70 oder gar 60 Zeichen aussieht.

```
↑↑↑↑↑↑↑↑↑↑↑↑↑↑ Seite 1 ↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑
Das ist das Programm 'Texteditor' fuer d
en CPC464. Wenn Ihnen auch die Eingabe d
er Texte ein klein wenig langsam vorkomm
t ... kein Problem: der CPC hat einen so
genannten Buffer fuer viele Zeichen.↓
↓
Mit diesem Pfeil trennen wir Absaetze.↓
↓
Wir koennen insgesamt 10 Bildschirmseite
n Text eintippen, nach etwas suchen lass
en, abspeichern, laden, drucken ...↓
↓
        und voll editieren!↓
↓
Viel Spass bei der Arbeit mit diesem Pro
gramm!↓
↓

↓↓↓↓↓↓↓↓↓↓↓↓↓↓ Seite 2 ↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
_21  1         <CTRL> + SLFDB
```

```
10 REM Texteditor
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                         Rainer Lueers
40 ON ERROR GOTO 850
50 MODE 1:DIM a$(10,23):seite=1:s1=1
60 FOR n=1 TO 10
70 FOR m=1 TO 23
80 a$(n,m)=STRING$(40," ")
90 NEXT m,n
100 REM Erstaufbau des Bildschirms
110 WINDOW #1,2,39,1,1
120 WINDOW #2,1,40,2,23
130 WINDOW #3,2,39,24,24
140 WINDOW #4,1,40,25,25
150 PAPER #1,3
160 PAPER #3,3
170 PAPER #4,3
180 CLS #1:CLS #2:CLS #3:CLS #4
190 PRINT #1,STRING$(38,CHR$(94));
200 PRINT #3,STRING$(38,CHR$(241));
210 CLS #4:PRINT #4,VPOS(#2);POS(#2)
220 LOCATE #4,15,1:PRINT #4,"<CTRL> + SL
FDB";:LOCATE #4,POS(#2),1:PRINT #4,CHR$(
22);CHR$(1);CHR$(95);CHR$(22);CHR$(0);
230 IF s1<seite THEN s1=seite
240 REM Warten auf Tastendruck
250 a$=INKEY$
260 IF a$="" THEN GOTO 250
270 GOTO 630
280 GOSUB 330
290 GOSUB 550
300 GOSUB 570
310 PRINT #2,a$;:IF a$=CHR$(241) THEN PR
INT #2
320 GOTO 210
330 REM Bildschirmeditor:
       Erkennen von Begrenzung,
       Umblaettermoeglichkeit und
       <DEL>-Taste
```

```
340 IF ASC(a$)<244 AND ASC(a$)>240 AND m
erker<>1 THEN merker=1:v=VPOS(#2):p=POS(
#2)
350 REM CHR$(224)
         heisst: 'COPY' gedrueckt
         bedeutet: Cursor wieder
         zum Ausgangspunkt
360 IF ASC(a$)=224 AND merker=1 THEN mer
ker=0:LOCATE #2,p,v:a$="":GOTO 540
370 IF POS(#2)=40 AND VPOS(#2)=22 AND a$
<>CHR$(13) AND ASC(a$)<128 THEN PRINT CH
R$(7);:LOCATE #2,1,22:RETURN
380 REM CHR$(127)
         heisst: 'DEL' gedrueckt
         bedeutet: Zeichen links vom
         Cursor wird geloescht
390 IF ASC(a$)=127 AND VPOS(#2)<>1 AND P
OS(#2)=1 THEN LOCATE #2,40,VPOS(#2)-1:PR
INT #2," ";:LOCATE #2,40,VPOS(#2)-1:a$="
":GOTO 540
400 IF ASC(a$)=127 AND POS(#2)<>1 THEN L
OCATE #2,POS(#2)-1,VPOS(#2):PRINT #2," "
;:LOCATE #2,POS(#2)-1,VPOS(#2):a$="":GOT
O 540
410 REM CHR$(240)
         heisst: Pfeil 'oben' gedrueckt
         bedeutet: Cursor eine Zeile bzw.
         eine Seite hoeher
420 IF ASC(a$)=240 AND VPOS(#2)<>1 THEN
LOCATE #2,POS(#2),VPOS(#2)-1:a$="":GOTO
540 ELSE IF ASC(a$)=240 AND VPOS(#2)=1 T
HEN IF seite<>1 THEN seite=seite-1:CLS #
2:FOR n=1 TO 22:PRINT #2,a$(seite,n);:NE
XT n:LOCATE #2,1,1
430 REM CHR$(241)
         heisst: Pfeil 'unten' gedrueckt
         bedeutet: Cursor eine Zeile bzw.
         eine Seite tiefer
```

```
440 IF ASC(a$)=241 AND VPOS(#2)<>22 THEN
 LOCATE #2,POS(#2),VPOS(#2)+1:a$="":GOTO
 540 ELSE IF ASC(a$)=241 AND VPOS(#2)=22
 THEN seite=seite+1:CLS #2:FOR n=1 TO 22
:PRINT #2,a$(seite,n);:NEXT n:LOCATE #2,
1,1
450 REM CHR$(242)
        heisst: Pfeil 'links'
        bedeutet: Cursor eine Position
        nach links
460 IF ASC(a$)=242 AND VPOS(#2)<>1 AND P
OS(#2)=1 THEN LOCATE #2,VPOS(#2)-1,40:a$
="":GOTO 540
470 IF ASC(a$)=242 AND VPOS(#2)<>1 THEN
LOCATE #2,POS(#2)-1,VPOS(#2):a$="":GOTO
540
480 REM CHR$(243)
        heisst: Pfeil 'rechts'
        bedeutet: Cursor eine Position
        nach rechts
490 IF ASC(a$)=243 AND VPOS(#2)<>22 AND
POS(#2)=40 THEN LOCATE #2,1,VPOS(#2)+1:a
$="":GOTO 540
500 IF ASC(a$)=243 AND VPOS(#2)<>22 THEN
 LOCATE #2,POS(#2)+1,VPOS(#2):a$="":GOTO
 540
510 IF ASC(a$)<244 AND ASC(a$)>239 THEN
merker=0:a$="":GOTO 540
520 IF ASC(a$)=224 THEN a$=""
530 IF a$=CHR$(13) THEN a$=CHR$(241)
540 RETURN
550 IF VPOS(#2)=22 AND a$=CHR$(241) THEN
 PRINT #2,a$;:MID$(a$(seite,VPOS(#2)),PO
S(#2)-1,1)=a$:seite=seite+1:a$="":CLS #2
560 RETURN
570 LOCATE #1,15,1:PRINT #1," Seite";sei
te;
580 LOCATE #3,15,1:PRINT #3," Seite";sei
te+1;
```

```
590 IF a$<>"" THEN MID$(a$(seite,VPOS(#2
)),POS(#2),1)=a$
600 RETURN
610 REM Ueberpruefen, welche Taste
        gedrueckt wurde
620 REM <CTRL> 'B'
        Blaettern
630 IF ASC(a$)=2 AND seite<>10 THEN seit
e=seite+1:CLS #2:FOR n=1 TO 22:PRINT #2,
a$(seite,n);:NEXT n:LOCATE #2,1,1:GOTO 3
00
640 REM <CTRL> 'S'
        Abspeichern auf Kassette
650 IF ASC(a$)=19 THEN CLS #4:INPUT #4,"
Name ";f$:IF f$="" THEN GOTO 820 ELSE f$
="!"+f$:OPENOUT f$:PRINT #9,s1:FOR n=1 T
O s1:FOR m=1 TO 22:PRINT #9,a$(n,m):NEXT
 m,n:CLOSEOUT:GOTO 710
660 REM <CTRL> 'L'
        Laden von Kassette
670 IF ASC(a$)=12 THEN CLS #4:INPUT #4,"
Name ";f$:IF f$="" THEN GOTO 820 ELSE f$
="!"+f$:OPENIN f$:INPUT #9,s1:FOR n=1 TO
 s1:FOR m=1 TO 22:INPUT #9,a$(n,m):NEXT
m,n:CLOSEIN:seite=1:CLS #2:FOR n=1 TO 22
:PRINT #2,a$(seite,n);:NEXT n:GOTO 300
680 REM <CTRL> 'F'
        Finden von Suchbegriffen im Text
690 IF ASC(a$)=6 THEN CLS #4:INPUT #4,"S
uchbegriff ";f$:IF f$="" THEN GOTO 710 E
LSE FOR n=1 TO s1:FOR m=1 TO 22:IF INSTR
(a$(n,m),f$)>0 THEN CLS #2:FOR mm=1 TO 2
2:PRINT #2,a$(n,mm);:NEXT mm:seite=n:LOC
ATE #2,1,m:GOTO 710 ELSE NEXT m,n
700 REM Wird der Suchbegriff gefunden,
        kann nach Aufforderung weiter-
        gesucht werden
```

```
710 IF ASC(a$)=6 THEN CLS#4:PRINT#4,n;m;
:INPUT#4,"Mehr(J/)";ff$:ff$=UPPER$(ff$):
IF ff$="J"THEN FOR n=seite TO s1:FOR m=V
POS(#2)+1 TO 23:IF INSTR(a$(n,m),f$)>0 T
HEN CLS#2:FOR mm=1 TO 22:PRINT#2,a$(n,mm
);:NEXT mm:seite=n:LOCATE#2,1,m:GOTO 710
 ELSE NEXT m,n
720 REM <CTRL> 'D'
         Drucker-/80-Zeichen-Routine
730 IF ASC(a$)=4 THEN CLS#4 ELSE GOTO 82
0
740 INPUT #4,"Drucker angestellt (J/ ) "
;ff$:ff$=UPPER$(ff$):IF LEFT$(ff$,1)<>"J
" THEN GOTO 820
750 CLS#4:INPUT #4,"Zeichen pro Zeile (>
=40) ";z1:IF z1<40 THEN GOTO 750
760 a$="":FOR z2=1 TO s1:FOR z3=1 TO 22:
FOR z4=1 TO 40
770 IF MID$(a$(z2,z3),z4,1)=" " THEN z5=
LEN(a$) ELSE IF MID$(a$(z2,z3),z4,1)=CHR
$(241) THEN z4=40:GOTO 790
780 a$=a$+MID$(a$(z2,z3),z4,1):IF LEN(a$
)=z1 THEN PRINT #8,LEFT$(a$,z5):a$=RIGHT
$(a$,LEN(a$)-z5):GOTO 800 ELSE GOTO 800
790 PRINT #8,a$:a$=""
800 IF LEFT$(a$,1)=" " THEN a$=RIGHT$(a$
,LEN(a$)-1)
810 NEXT z4,z3,z2:MODE 1:GOTO 100
820 IF ASC(a$)<32 AND ASC(a$)<>13 THEN a
$=""
830 GOTO 280
840 REM Errorbehandlungsroutine
850 a$=" ":RESUME NEXT
860 a$=a$+MID$(a$(z2,z3),z4,1):IF (LEN(a
$)=z1 AND MID$(a$(z2,z3),z4,1)<>" ") THE
N PRINT LEFT$(a$,z5):a$=RIGHT$(a$,LEN(a$
)-z5):GOTO 800 ELSE IF RIGHT$(a$,1)=CHR$
(241) THEN a$=LEFT$(a$,LEN(a$)-1):z4=40
 ELSE GOTO 800
```

```
Das ist das Programm 'Texteditor' fuer den CPC464. Wenn Ihnen auch die Eingabe
der Texte ein klein wenig langsam vorkommt ... kein Problem: der CPC hat einen
sogenannten Buffer fuer viele Zeichen.

Mit diesem Pfeil trennen wir Absaetze.

Wir koennen insgesamt 10 Bildschirmseiten Text eintippen, nach etwas suchen
lassen, abspeichern, laden, drucken ...

und voll editieren!

Viel Spass bei der Arbeit mit diesem Programm!

Ready
```

Deutscher Zeichensatz

Alle schimpfen sie darüber, wenn sie Ihren modernen CPC neben der alt-ehrwürdigen Schreibmaschine stehen sehen: Auf der einen Seite ein internationaler Computer ... natürlich auch mit internationalem Zeichensatz (also ohne deutsche Umlaute und 'sz'), auf der anderen Seite die Schreibmaschine, die über all' diesen nützlichen Komfort verfügt, dafür aber halt (normalerweise) kein Computer ist.

Mit diesem Programm machen wir aus Ihrem CPC ein Pendant zur Schreibmaschine zumindest was die Tastatur angeht: das 'z' sitzt dann da, wo es sitzen soll, nämlich beim 'y' - und entsprechend umgekehrt. Die Umlaute 'ae', 'ue', 'oe' sowie das 'sz' finden sich nun auch auf der Tastatur mehr oder weniger am gleichen Ort wie auf der Schreibmaschine.

Nun werden Sie aber fragen, was denn mit den Tastaturbeschriftungen passiert. Die bleiben vorerst natürlich so, wie sie schon immer waren, es sei denn, Sie kleben kleine Zettelchen auf die entsprechenden Tastaturoberseiten.

Und was geschieht mit den alten Zeichen, die eigentlich auf den neu belegten Tasten zu finden sind? Im Programmlisting ist die jeweilige Zuordnung angegeben (ab Zeile 180). Zum großen Teil wird die alte Tastaturbelegung - die internationale - durch gleichzeitiges Drücken mit der (CTRL)-Taste erzeugt.

Sind Sie den deutschen Zeichensatz leid und wollen wieder auf der internationalen Tastatur hämmern? Entweder Computer ausstellen oder ... von der Zehnertastatur die Taste '0' drücken -) wieder die alte = internationale Tastaturbelegung; später einfach die '1' von der Zehnertastatur gedrückt -) wieder die neue = deutsche Tastaturbelegung! Diese Umschalterei funktioniert allerdings nur, wenn Sie das Programm 'Deutscher Zeichensatz' noch im Speicher haben. Vorsicht bei Anwendung von 'RENUM' zum Umnummerieren! Die Funktionstasten '1' und '0' (Belegung siehe Zeile 120 und 130) müssen dann per Hand umbelegt oder das Programm dementsprechend umgeschrieben werden.

```
10 REM Deutscher Zeichensatz
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                            Rainer Lueers
40 SYMBOL AFTER 32
50 REM Einlesen des deutschen Zeichen-
       satzes aus den DATA-Zeilen
60 FOR m=1 TO 7
70 READ a
80 FOR n=1 TO 8
90 READ a(n)
100 NEXT n
110 SYMBOL a,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
120 NEXT m
130 REM Funktionstastenbelegung mit
        Zeilennummer zur Aktivierung
        des alten (0) bzw. neuen (1)
        Zeichensatzes
140 KEY 1,"GOTO 160"+CHR$(13)
150 KEY 0,"GOTO 400"+CHR$(13)
160 REM Neuer Zeichensatz DEUTSCH
170 '   =========================
180 REM z->y Z->Y CTRL z->z
190 KEY DEF 71,1,121,89,122
200 REM y->z Y->Z CTRL y->y
210 KEY DEF 43,1,122,90,121
220 REM ;->ä +->Ä CTRL ;->;
230 KEY DEF 28,1,123,91,59
240 REM §->ü "ö"->ü
250 KEY DEF 26,1,125,93
260 REM :->"ö" *->ö
270 KEY DEF 29,1,124,92
280 REM ↑->"ß" "#"->↑ CTRL ↑->"#"
290 KEY DEF 24,1,126,94,163
300 REM Ä->: ä->* CTRL Ä->Ä
310 KEY DEF 17,1,58,42,91
320 REM Ü->; ü->+ CTRL Ü-Ü
330 KEY DEF 19,1,59,43,93
```

```
340 REM ö->§
350 KEY DEF 22,1,64
360 MODE 1
370 REM Beispielsatz mit saemtlichen
        Umlauten und sz
380 PRINT"über den großen Fluß gehen zu
dürfen,   das bedeutet für mich ein Ärge
rnis      ohnegleichen ... ich rufe laut
 'RöMER'  ... mmh ... ein löbliches Wort
 auf ...  ääh ... jetzt fällt mir nichts
 mehr ein."
390 END
400 REM Alter Zeichensatz INTERNAT.
410 '   ===========================
420 REM y->z Y->Z CTRL z->""
430 KEY DEF 71,1,122,90,26
440 REM z->y Z->Y CTRL y->""
450 KEY DEF 43,1,121,89,25
460 REM ä->; Ä->+
470 KEY DEF 28,1,59,43
480 REM ü->§ Ü->"ö"
490 KEY DEF 26,1,64,124
500 REM "ö">: ö->*
510 KEY DEF 29,1,58,42
520 REM "ß"->↑ ↑->"#" CTRL "#"->""
530 KEY DEF 24,1,94,163,30
540 REM :->Ä *->ä CTRL Ä->"
550 KEY DEF 17,1,91,123,27
560 REM ;->Ü +->ü CTRL Ü->""
570 KEY DEF 19,1,93,125,29
580 REM §->ö
590 KEY DEF 22,1,92
600 END
610 REM DATA-Zeilen mit deutschen
        Umlauten in Klein- und
        Grossschrift und dem sz
```

```
620 REM der Buchstabe 'sz'
630 DATA 126
640 DATA &x 01111000
650 DATA &x 11001100
660 DATA &x 11001100
670 DATA &x 11111000
680 DATA &x 11001100
690 DATA &x 11001100
700 DATA &x 11111000
710 DATA &x 11000000
720 REM der Buchstabe 'ae'
730 DATA 123
740 DATA &x 01101100
750 DATA &x 00000000
760 DATA &x 01111000
770 DATA &x 00001100
780 DATA &x 01111100
790 DATA &x 11001100
800 DATA &x 01110110
810 DATA &x 00000000
820 REM der Buchstabe 'oe'
830 DATA 124
840 DATA &x 01101100
850 DATA &x 00000000
860 DATA &x 00111100
870 DATA &x 01100110
880 DATA &x 01100110
890 DATA &x 01100110
900 DATA &x 00111100
910 DATA &x 00000000
920 REM der Buchstabe 'ue'
930 DATA 125
940 DATA &x 01101100
950 DATA &x 00000000
960 DATA &x 01100110
970 DATA &x 01100110
980 DATA &x 01100110
990 DATA &x 01100110
1000 DATA &x 00111111
1010 DATA &x 00000000
```

```
1020 REM der Buchstabe 'AE'
1030 DATA 91
1040 DATA &x 01101100
1050 DATA &x 00000000
1060 DATA &x 00011000
1070 DATA &x 00111100
1080 DATA &x 01100110
1090 DATA &x 01111110
1100 DATA &x 01100110
1110 DATA &x 00000000
1120 REM der Buchstabe 'OE'
1130 DATA 92
1140 DATA &x 01101100
1150 DATA &x 00000000
1160 DATA &x 01111100
1170 DATA &x 11000110
1180 DATA &x 11000110
1190 DATA &x 11000110
1200 DATA &x 01111100
1210 DATA &x 00000000
1220 REM der Buchstabe 'UE'
1230 DATA 93
1240 DATA &x 01101100
1250 DATA &x 00000000
1260 DATA &x 01100110
1270 DATA &x 01100110
1280 DATA &x 01100110
1290 DATA &x 01100110
1300 DATA &x 00111100
1310 DATA &x 00000000
```

```
über den großen Fluß gehen zu dürfen,
das bedeutet für mich ein Ärgernis
ohnegleichen ... ich rufe laut 'RÖMER'
... mmh ... ein löbliches Wort auf ...
ääh ... jetzt fällt mir nichts mehr ein.

Ready

█
```

Mathematikzeichensatz

Dieses Programm gliedert sich lediglich in zwei Abschnitte:

In Zeile 410 bis Zeile 2560 sind in DATA-Zeilen 24 häufig vorkommende mathematische Zeichen samt ihrer Bedeutung aufgeführt. Im einzelnen sind es folgende Zeichen, die auf der Tastatur durch (CTRL) und eine Buchstabentaste angesprochen werden:

Bitte (CTRL) und die entsprechend angegebene Buchstabentaste zusammen drücken:

Alpha	a	Beta	b	Gamma	c
nicht El.v.	d	Element v.	e	Integral	f
Pi	g	Sigma	h	Omega	i
unendlich	j	Winkel	k	Dreieck	l
Viereck	n	Kreis	o	parall.	q
rechtw. zu	r	kongruent	s	aehnlich	t
leere Menge	u	1 zu 1	v	ungleich	w
angenähert	x	proportional	y	angen.an	z

Wir haben hierzu die binäre Darstellungsform gewählt, um Ihnen die Möglichkeit zu eröffnen, die DATA-Zeilen nach eigenem Gutdünken leicht zu verändern. Hierzu ein Hinweis: Sie dürfen pro Zeile nur acht Zeichen vergeben (jeweils Nullen oder Einsen; null bedeutet Pixel nicht gesetzt, eins bedeutet Pixel gesetzt).

Jedes Zeichen muß genau aus acht Zeilen zu je acht Zeichen zusammengesetzt sein. Wird irgendwo nur eine DATA-Zeile zusätzlich eingefügt, so verändert sich der nachfolgende Zeichensatz vollständig (probieren Sie es ruhig einmal aus!).

In Zeile 80 bis 390 findet das Einlesen der DATAs durch 'READ' statt. Um die binäre Darstellung auch richtig als Zahlwert zu deuten, muß die '01'-Darstellung als String eingelesen werden und vor der Umwandlung in eine zu verarbeitende Dezimalzahl ('VAL') die Kennzeichnung '&x' = Binärzahl vorangestellt werden.

Nach dem Starten des Programms mit 'RUN' können die Mathematikzeichen durch Drücken eines Buchstabens (siehe Tabelle oben) zusammen mit der Taste (CTRL) auf dem Bildschirm dargestellt werden.

Da der normale ASCII-Zeichensatz auch weiterhin gebraucht werden soll, haben wir die als Zeichen sonst nicht darstellbaren (CTRL)-Codes 1 bis 26 mit den Mathematikzeichen belegt. ASCII-Zeichen 13 ließ sich nicht umändern, weil dieser Character bereits mit der Bedeutung 'Cursor zur ersten Schreibstelle' vorbelegt war (ENTER)-Taste drücken. Auch ASCII-Zeichen 16 blieb ausgespart, da diese Taste bereits die Bedeutung hat: 'Zeichen auf Cursorposition löschen'.



```
10 REM Mathematikzeichensatz
20 REM CPC Basic Programme
30 REM Copyright DATA BECKER &
                    Rainer Lueers
40 RESTORE
50 SYMBOL AFTER 1
60 REM Neubelegen der
       ASCII-Zeichen 1 bis 12
70 REM Ansprechbar durch
       <CTRL> und Buchstabe
80 FOR m=1 TO 12
90 FOR n=1 TO 8
100 READ a$
110 REM Umbenennen des eingelesenen
        Strings in eine Binaer- und
        dann eine Dezimalzahl
120 a$(n)="&x "+a$
130 a(n)=VAL("&x "+a$)
140 NEXT n
150 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
160 NEXT m
170 REM Neubelegen der
        ASCII-Zeichen 14 und 15
180 REM Ansprechen durch
        <CTRL> und Buchstabe
190 FOR m=14 TO 15
200 FOR n=1 TO 8
210 READ a$
220 REM Umbenennen des eingelesenen
        Strings in eine Binaer- und
        dann eine Dezimalzahl
230 a$(n)="&x "+a$
240 a(n)=VAL("&x "+a$)
250 NEXT n
260 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
270 NEXT m
280 REM Neubelegung der
        ASCII-Zeichen 17 bis 26
```

```
290 REM Ansprechen durch
        <CTRL> und Buchstabe
300 FOR m=17 TO 26
310 FOR n=1 TO 8
320 READ a$
330 REM Umbenennen des eingelesenen
        Strings in eine Binaer- und
        dann eine Dezimalzahl
340 a$(n)="&x "+a$
350 a(n)=VAL("&x "+a$)
360 NEXT n
370 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
380 NEXT m
390 END
400 REM Es folgt die binaere Darstellung
        des Mathematikzeichensatzes.
        Vorangestellt ist jeweils in
        einer REM-Zeile die Bedeutung
        des Zeichens
410 REM Alpha
420 DATA 00000000
430 DATA 00011100
440 DATA 00100100
450 DATA 01000100
460 DATA 01000100
470 DATA 00100100
480 DATA 00011100
490 DATA 00000010
500 REM Beta
510 DATA 00000000
520 DATA 01111000
530 DATA 01000100
540 DATA 01011000
550 DATA 01000100
560 DATA 01000100
570 DATA 01111000
580 DATA 10000000
```

```
590 REM Gamma
600 DATA 00000000
610 DATA 10000010
620 DATA 01000100
630 DATA 00101000
640 DATA 00010000
650 DATA 00010000
660 DATA 00010000
670 DATA 00010000
680 REM nicht Element von
690 DATA 00000000
700 DATA 00000100
710 DATA 00011110
720 DATA 00100100
730 DATA 00111110
740 DATA 00100100
750 DATA 00011110
760 DATA 00000100
770 REM Element von
780 DATA 00000000
790 DATA 00000000
800 DATA 00011110
810 DATA 00100000
820 DATA 00111110
830 DATA 00100000
840 DATA 00011110
850 DATA 00000000
860 REM Integralzeichen
870 DATA 00110000
880 DATA 00101000
890 DATA 00101000
900 DATA 00100000
910 DATA 00100000
920 DATA 10100000
930 DATA 10100000
940 DATA 01100000
```

```
950 REM Pi
960 DATA 00000000
970 DATA 00000000
980 DATA 01111110
990 DATA 00100100
1000 DATA 00100100
1010 DATA 00100100
1020 DATA 00100100
1030 DATA 00000000
1040 REM Sigma
1050 DATA 00000000
1060 DATA 01111111
1070 DATA 00100001
1080 DATA 00010000
1090 DATA 00010000
1100 DATA 00100001
1110 DATA 01111111
1120 DATA 00000000
1130 REM Omega
1140 DATA 00000000
1150 DATA 00111100
1160 DATA 00100100
1170 DATA 01000010
1180 DATA 01000010
1190 DATA 00100100
1200 DATA 11100111
1210 DATA 00000000
1220 REM unendlich
1230 DATA 00000000
1240 DATA 00000000
1250 DATA 01101100
1260 DATA 10010010
1270 DATA 10010010
1280 DATA 10010010
1290 DATA 01101100
1300 DATA 00000000
1310 DATA 00000000
```

```
1320 REM Winkel
1330 DATA 00000010
1340 DATA 00000100
1350 DATA 00001000
1360 DATA 00010000
1370 DATA 00100000
1380 DATA 01000000
1390 DATA 11111110
1400 REM Dreieck
1410 DATA 00000000
1420 DATA 00000000
1430 DATA 00000000
1440 DATA 00010000
1450 DATA 00101000
1460 DATA 01000100
1470 DATA 11111110
1480 DATA 00000000
1490 REM Viereck
1500 DATA 11111110
1510 DATA 10000010
1520 DATA 10000010
1530 DATA 10000010
1540 DATA 10000010
1550 DATA 10000010
1560 DATA 11111110
1570 DATA 00000000
1580 REM Kreis
1590 DATA 00000000
1600 DATA 01111100
1610 DATA 10000010
1620 DATA 10000010
1630 DATA 10000010
1640 DATA 10000010
1650 DATA 01111100
1660 DATA 00000000
```

```
1670 REM parallel zu
1680 DATA 01001000
1690 DATA 01001000
1700 DATA 01001000
1710 DATA 01001000
1720 DATA 01001000
1730 DATA 01001000
1740 DATA 01001000
1750 DATA 01001000
1760 REM rechtwinklig zu
1770 DATA 00010000
1780 DATA 00010000
1790 DATA 00010000
1800 DATA 00010000
1810 DATA 00010000
1820 DATA 00010000
1830 DATA 11111110
1840 DATA 00000000
1850 REM kongruent
1860 DATA 01100000
1870 DATA 10011001
1880 DATA 00000110
1890 DATA 00000000
1900 DATA 11111111
1910 DATA 00000000
1920 DATA 11111111
1930 DATA 00000000
1940 REM aehnlich, proportional
1950 DATA 00000000
1960 DATA 00000000
1970 DATA 00000000
1980 DATA 01100000
1990 DATA 10011001
2000 DATA 00000110
2010 DATA 00000000
2020 DATA 00000000
```

```
2030 REM leere Menge
2040 DATA 00000010
2050 DATA 01111100
2060 DATA 10001010
2070 DATA 10010010
2080 DATA 10100010
2090 DATA 01111100
2100 DATA 10000000
2110 DATA 00000000
2120 REM 1 zu 1
2130 DATA 00000000
2140 DATA 00100100
2150 DATA 01000010
2160 DATA 11111111
2170 DATA 01000010
2180 DATA 00100100
2190 DATA 00000000
2200 DATA 00000000
2210 REM nicht gleich, ungleich
2220 DATA 00000001
2230 DATA 00000010
2240 DATA 11111111
2250 DATA 00001000
2260 DATA 11111111
2270 DATA 00100000
2280 DATA 01000000
2290 DATA 00000000
2300 REM angenaehert, nahezu gleich
2310 DATA 00000000
2320 DATA 01100000
2330 DATA 10011001
2340 DATA 00000110
2350 DATA 01100000
2360 DATA 10011001
2370 DATA 00000110
2380 DATA 00000000
```

```
2390 REM proportional zu
2400 DATA 00000000
2410 DATA 00000000
2420 DATA 01110111
2430 DATA 10001000
2440 DATA 10001000
2450 DATA 10001000
2460 DATA 01110111
2470 DATA 00000000
2480 REM angenaehert an Wert ...
2490 DATA 00000000
2500 DATA 00000100
2510 DATA 00000010
2520 DATA 11111111
2530 DATA 00000010
2540 DATA 00000100
2550 DATA 00000000
2560 DATA 00000000
```

Computerzeichensatz

Ehrlich eingestanden kann man feststellen, daß der Zeichensatz des CPC sehr schön aussieht. Obgleich die Zeichen wie auch bei anderen Computern nur in einer 8*8 - Matrix erstellt sind (jeweils 8 horizontal und 8 vertikal gesetzte oder nicht gesetzte Pixel), muten sie mit der gewählten Detailtreue (schaut man sich z.B. die '4' oder den Buchstaben 'n' an) schreibmaschinenähnlich an.

Warum jetzt noch einen anderen Zeichensatz? Seit Jahren hält die Euphorie für Computerspiele an; warum soll für dieses futuristische Etwas der normale Zeichensatz herhalten - her mit dem entsprechenden Computerzeichensatz!

Das Programm gliedert sich in zwei Hauptteile:
1) Ein fast unübersehbares DATA-Zeichenmeer mit dem Computerzeichensatz
2) Die Verarbeitungs- und Laderoutine für die DATAS.

Auf Punkt 2 wollen wir etwas näher eingehen: Der CPC stellt alles (auch Buchstaben und andere Zeichen) auf dem Bildschirm grafisch dar. So ist es im Gegensatz zu anderen Computern auch verhältnismäßig leicht möglich, nicht nur Text mit Grafik zu mischen, nein auch dasselbe Charakterzeichen mehrfach in unterschiedlichem Aussehen auf der Bildschirmseite abzubilden. Dieser Zweck wurde im Programm 'Computerzeichensatz' folgendermaßen ausgenutzt: Es kann einerseits der gesamte Zeichensatz in Computerschrift umgewandelt werden, es kann andererseits mit Normal- und Computerzeichensatz gleichzeitig gearbeitet werden.

Mit dem komfortablen SYMBOL-Befehl dürfte Möglichkeit eins klar sein.

Wie werden aber zwei Zeichensätze gleichzeitig im Speicher gehalten? Hierfür ist es notwendig, daß das Programm weiterläuft, denn so viele verschiedene Zeichen kann man nicht so einfach gleichzeitig mit der Tastatur ansprechen. Dafür haben wir nun bei laufendem Programm einen Schalter zwischen den Zeichensätzen

eingebaut: (TAB) gedrückt -) Computerzeichensatz an, (TAB) wieder gedrückt -) Normalzeichensatz an usw.

Programmtechnisch spielt sich folgendes ab: Der ASCII-Code des gewünschten Zeichens wird interpretiert (z.B. '!' = 33); dazu wird das entsprechende Computerzeichensatzild gesucht und ... nicht mit 'SYMBOL' über unseren gewählten Buchstaben in Normalschrift gelegt (dann wäre dieser nämlich fortan Computerzeichen!) ... in ein nicht benötigtes Zeichen ('CHR$(255)') geladen. Letzteres Zeichen wird daraufhin immer und immer wieder, jedoch in dem gewünschten Aussehen, als Computerschriftzeichen zur Abbildung gebracht.



```
10 REM Computerschrift
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                        Rainer Lueers
40 MODE 1
50 DIM a(122,8)
60 REM Umdefinierbaren Zeichenraum
       freigeben
70 REM Einlesen der Computerschrift in
       die dimensionierte Variable a
80 SYMBOL AFTER 32
90 REM Zeichencode und Zeichenausehen
       vorlaeufig in die dimensionierte
       Variable 'a' laden
100 FOR n=1 TO 73
110 READ a
120 FOR m=1 TO 8
130 READ a$
140 REM Umwandlung von der Stringdar-
        stellung in eine Binaerzahl und
        anschliessend in eine Dezimal-
        zahl
150 a$="&x "+a$
160 a(a,m)=VAL(a$)
170 NEXT m
180 NEXT n
190 REM Auswahlmoeglichkeiten:
        1) bei laufendem Programm
           Text in Computerschrift oder
           Normalschrift einzugeben
        2) den ganzen Zeichensatz in
           Computerschrift umzuwandeln
200 PRINT "Bitte geben Sie etwas ein! So
ll es in  'Computerschrift' erscheinen,
 so druek- ken Sie bitte vorher die <TAB
>-Taste.  Soll in Normalschrift weiterg
eschrie-  ben werden, so druecken Sie b
itte wie- der die <TAB>-Taste. Sollen a
lle"
```

```
210 PRINT "Zeichen von nun an (auch das
Listing)    in Computerschrift erscheinen
, so        druecken Sie bitte jetzt die
Taste j";:a$="":INPUT a$
220 a$=UPPER$(a$):IF a$="J" THEN GOTO 51
Ø
23Ø REM Der Normalzeichensatz bleibt
        erhalten, nur einzelne Buchsta-
        ben werden auf Wunsch in Com-
        puterzeichen umgewandelt
24Ø PRINT
25Ø PRINT "Bitte jetzt Text eingeben"
26Ø PRINT
27Ø a$=INKEY$
28Ø IF a$="" THEN GOTO 27Ø
29Ø REM Kleiner Texteditor: Die Tasten
        <ENTER>=13, <TAB>=9 und <DELETE>
        =127 werden interpretiert
3ØØ IF ASC(a$)=9 THEN GOTO 36Ø
31Ø IF ASC(a$)=13 THEN PRINT:GOTO 27Ø
32Ø IF ASC(a$)=127 THEN a$="":IF POS(#Ø)
>1 THEN LOCATE POS(#Ø)-1,VPOS(#Ø):PRINT
" ";:LOCATE POS(#Ø)-1,VPOS(#Ø):GOTO 27Ø
33Ø PRINT a$;
34Ø GOTO 27Ø
35Ø END
36Ø a$=INKEY$
37Ø IF a$="" THEN GOTO 36Ø
38Ø REM Kleiner Texteditor: Die Tasten
        <ENTER>=13, <TAB>=9 und <DELETE>
        =127 werden interpretiert
39Ø IF ASC(a$)=9 THEN GOTO 27Ø
4ØØ IF ASC(a$)=13 THEN PRINT:GOTO 36Ø
41Ø IF ASC(a$)=127 THEN a$="":IF POS(#Ø)
>1 THEN LOCATE POS(#Ø)-1,VPOS(#Ø):PRINT
" ";:LOCATE POS(#Ø)-1,VPOS(#Ø):GOTO 36Ø
ELSE GOTO 36Ø
42Ø IF ASC(a$)>122 THEN PRINT a$;:GOTO 3
6Ø
```

```
430 IF a(ASC(a$),1)=0 AND a(ASC(a$),2)=0
 AND a(ASC(a$),3)=0 AND a(ASC(a$),4)=0 A
ND a(ASC(a$),5)=0 AND a(ASC(a$),6)=0 AND
 a(ASC(a$),7)=0 AND a(ASC(a$),8)=0 THEN
PRINT a$;:GOTO 360
440 REM Damit der normale Zeichensatz
        nicht zerstoert wird, wird jedes
        Zeichen als ein umgewandelter
        CHR$(255) dargestellt
450 z=ASC(a$)
460 SYMBOL 255,a(z,1),a(z,2),a(z,3),a(z,
4),a(z,5),a(z,6),a(z,7),a(z,8)
470 PRINT CHR$(255);
480 GOTO 360
490 END
500 REM Einlesen der Computerschrift und
        direkte Veraenderung des
        normalen Zeichensatzes
510 RESTORE
520 FOR n=1 TO 73
530 READ m
540 FOR o=1 TO 8
550 READ p$
560 REM Umwandlung von der Stringdar-
        stellung in eine Binaerzahl und
        anschliessend in eine Dezimal-
        zahl
570 p$="&x "+p$
580 b(o)=VAL(p$)
590 NEXT o
600 SYMBOL m,b(1),b(2),b(3),b(4),b(5),b(
6),b(7),b(8)
610 NEXT n
620 END
```

```
63Ø REM !
64Ø DATA 33
65Ø DATA ØØ111ØØØ
66Ø DATA ØØ111ØØØ
67Ø DATA ØØ111ØØØ
68Ø DATA ØØ111ØØØ
69Ø DATA ØØØ11ØØØ
7ØØ DATA ØØØØØØØØ
71Ø DATA ØØØ11ØØØ
72Ø DATA ØØØØØØØØ
73Ø REM "
74Ø DATA 34
75Ø DATA Ø11ØØ11Ø
76Ø DATA Ø11ØØ11Ø
77Ø DATA Ø11ØØ11Ø
78Ø DATA ØØØØØØØØ
79Ø DATA ØØØØØØØØ
8ØØ DATA ØØØØØØØØ
81Ø DATA ØØØØØØØØ
82Ø DATA ØØØØØØØØ
83Ø REM #
84Ø DATA 35
85Ø DATA Ø11ØØ11Ø
86Ø DATA 11111111
87Ø DATA Ø11ØØ11Ø
88Ø DATA Ø11ØØ11Ø
89Ø DATA 11111111
9ØØ DATA Ø11ØØ11Ø
91Ø DATA ØØØØØØØØ
92Ø DATA ØØØØØØØØ
93Ø  REM '
94Ø  DATA 39
95Ø  DATA ØØØ11ØØØ
96Ø  DATA ØØØ11ØØØ
97Ø  DATA ØØØ11ØØØ
98Ø  DATA ØØØØØØØØ
99Ø  DATA ØØØØØØØØ
1ØØØ DATA ØØØØØØØØ
1Ø1Ø DATA ØØØØØØØØ
1Ø2Ø DATA ØØØØØØØØ
```

```
1030 REM (
1040 DATA 40
1050 DATA 00011110
1060 DATA 00011000
1070 DATA 00011000
1080 DATA 00111000
1090 DATA 00111000
1100 DATA 00111000
1110 DATA 00111110
1120 DATA 00000000
1130 REM )
1140 DATA 41
1150 DATA 01111000
1160 DATA 00011000
1170 DATA 00011000
1180 DATA 00011100
1190 DATA 00011100
1200 DATA 00011100
1210 DATA 01111100
1220 DATA 00000000
1230 REM 0
1240 DATA 48
1250 DATA 01111111
1260 DATA 01100011
1270 DATA 01100011
1280 DATA 01100011
1290 DATA 01100011
1300 DATA 01100011
1310 DATA 01111111
1320 DATA 00000000
1330 REM 1
1340 DATA 49
1350 DATA 00111000
1360 DATA 00011000
1370 DATA 00011000
1380 DATA 00011000
1390 DATA 00111110
1400 DATA 00111110
1410 DATA 00111110
1420 DATA 00000000
```

```
1430 REM 2
1440 DATA 50
1450 DATA 01111111
1460 DATA 00000011
1470 DATA 00000011
1480 DATA 01111111
1490 DATA 01100000
1500 DATA 01100000
1510 DATA 01111111
1520 DATA 00000000
1530 REM 3
1540 DATA 51
1550 DATA 01111110
1560 DATA 00000110
1570 DATA 00000110
1580 DATA 01111111
1590 DATA 00000111
1600 DATA 00000111
1610 DATA 11111111
1620 DATA 00000000
1630 REM 4
1640 DATA 52
1650 DATA 01110000
1660 DATA 01110000
1670 DATA 01110000
1680 DATA 01110111
1690 DATA 01110111
1700 DATA 01111111
1710 DATA 00000111
1720 DATA 00000000
1730 REM 5
1740 DATA 53
1750 DATA 01111111
1760 DATA 01100000
1770 DATA 01100000
1780 DATA 01111111
1790 DATA 00000111
1800 DATA 00000111
1810 DATA 01111111
1820 DATA 00000000
```

```
1830 REM 6
1840 DATA 54
1850 DATA 01111100
1860 DATA 01101100
1870 DATA 01100000
1880 DATA 01111111
1890 DATA 01100011
1900 DATA 01100011
1910 DATA 01111111
1920 DATA 00000000
1930 REM 7
1940 DATA 55
1950 DATA 01111111
1960 DATA 00000011
1970 DATA 00000011
1980 DATA 00011111
1990 DATA 00011000
2000 DATA 00011000
2010 DATA 00011000
2020 DATA 00000000
2030 REM 8
2040 DATA 56
2050 DATA 00111110
2060 DATA 00110110
2070 DATA 00110110
2080 DATA 01111111
2090 DATA 01110111
2100 DATA 01110111
2110 DATA 01111111
2120 DATA 00000000
2130 REM 9
2140 DATA 57
2150 DATA 01111111
2160 DATA 01100011
2170 DATA 01100011
2180 DATA 01111111
2190 DATA 00000111
2200 DATA 00000111
2210 DATA 00000111
2220 DATA 00000000
```

```
2230 REM :
2240 DATA 58
2250 DATA 00000000
2260 DATA 00011000
2270 DATA 00011000
2280 DATA 00000000
2290 DATA 00011000
2300 DATA 00011000
2310 DATA 00000000
2320 DATA 00000000
2330 REM ;
2340 DATA 59
2350 DATA 00000000
2360 DATA 00011000
2370 DATA 00011000
2380 DATA 00000000
2390 DATA 00011000
2400 DATA 00011000
2410 DATA 00110000
2420 DATA 00000000
2430 REM =
2440 DATA 61
2450 DATA 00000000
2460 DATA 01111110
2470 DATA 00000000
2480 DATA 00000000
2490 DATA 01111110
2500 DATA 00000000
2510 DATA 00000000
2520 DATA 00000000
2530 REM ?
2540 DATA 63
2550 DATA 01111111
2560 DATA 01100011
2570 DATA 00000011
2580 DATA 00011111
2590 DATA 00011100
2600 DATA 00000000
2610 DATA 00011100
2620 DATA 00000000
```

```
2630 REM §
2640 DATA 64
2650 DATA Ø1111111
2660 DATA Ø11ØØØ11
2670 DATA Ø11Ø1111
2680 DATA Ø11Ø1111
2690 DATA Ø11Ø1111
2700 DATA Ø11ØØØØØ
2710 DATA Ø1111111
2720 DATA ØØØØØØØØ
2730 REM A
2740 DATA 65
2750 DATA ØØ111111
2760 DATA ØØ11ØØ11
2770 DATA ØØ11ØØ11
2780 DATA Ø1111111
2790 DATA Ø111ØØ11
2800 DATA Ø111ØØ11
2810 DATA Ø111ØØ11
2820 DATA ØØØØØØØØ
2830 REM B
2840 DATA 66
2850 DATA Ø111111Ø
2860 DATA Ø11ØØ11Ø
2870 DATA Ø11ØØ11Ø
2880 DATA Ø1111111
2890 DATA Ø11ØØ111
2900 DATA Ø11ØØ111
2910 DATA Ø1111111
2920 DATA ØØØØØØØØ
2930 REM C
2940 DATA 67
2950 DATA Ø1111111
2960 DATA Ø11ØØ111
2970 DATA Ø11ØØ111
2980 DATA Ø11ØØØØØ
2990 DATA Ø11ØØØ11
3000 DATA Ø11ØØØ11
3010 DATA Ø1111111
3020 DATA ØØØØØØØØ
```

```
3030 REM D
3040 DATA 68
3050 DATA 01111110
3060 DATA 01100110
3070 DATA 01100110
3080 DATA 01110111
3090 DATA 01110111
3100 DATA 01110111
3110 DATA 01111111
3120 DATA 00000000
3130 REM E
3140 DATA 69
3150 DATA 01111111
3160 DATA 01100000
3170 DATA 01100000
3180 DATA 01111111
3190 DATA 01110000
3200 DATA 01110000
3210 DATA 01111111
3220 DATA 00000000
3230 REM F
3240 DATA 70
3250 DATA 01111111
3260 DATA 01100000
3270 DATA 01100000
3280 DATA 01111111
3290 DATA 01110000
3300 DATA 01110000
3310 DATA 01110000
3320 DATA 00000000
3330 REM G
3340 DATA 71
3350 DATA 01111111
3360 DATA 01100011
3370 DATA 01100000
3380 DATA 01101111
3390 DATA 01100111
3400 DATA 01100111
3410 DATA 01111111
3420 DATA 00000000
```

```
343Ø REM H
344Ø DATA 72
345Ø DATA Ø111ØØ11
346Ø DATA Ø111ØØ11
347Ø DATA Ø111ØØ11
348Ø DATA Ø1111111
349Ø DATA Ø111ØØ11
35ØØ DATA Ø111ØØ11
351Ø DATA Ø111ØØ11
352Ø DATA ØØØØØØØØ
353Ø REM I
354Ø DATA 73
355Ø DATA ØØØØ11ØØ
356Ø DATA ØØØØ11ØØ
357Ø DATA ØØØØ11ØØ
358Ø DATA ØØØØ11ØØ
359Ø DATA ØØ1111ØØ
36ØØ DATA ØØ1111ØØ
361Ø DATA ØØ1111ØØ
362Ø DATA ØØØØØØØØ
363Ø REM J
364Ø DATA 74
365Ø DATA ØØØØ11ØØ
366Ø DATA ØØØØ11ØØ
367Ø DATA ØØØØ11ØØ
368Ø DATA ØØØØ111Ø
369Ø DATA ØØØØ111Ø
37ØØ DATA Ø11Ø111Ø
371Ø DATA Ø111111Ø
372Ø DATA ØØØØØØØØ
373Ø REM K
374Ø DATA 75
375Ø DATA Ø11ØØ11Ø
376Ø DATA Ø11ØØ11Ø
377Ø DATA Ø11Ø11ØØ
378Ø DATA Ø1111111
379Ø DATA Ø11ØØ111
38ØØ DATA Ø11ØØ111
381Ø DATA Ø11ØØ111
382Ø DATA ØØØØØØØØ
```

```
3830 REM L
3840 DATA 76
3850 DATA ØØ11ØØØØ
3860 DATA ØØ11ØØØØ
3870 DATA ØØ11ØØØØ
3880 DATA Ø111ØØØØ
3890 DATA Ø111ØØØØ
3900 DATA Ø111ØØØØ
3910 DATA Ø111111Ø
3920 DATA ØØØØØØØØ
3930 REM M
3940 DATA 77
3950 DATA Ø11ØØ111
3960 DATA Ø1111111
3970 DATA Ø1111111
3980 DATA Ø111Ø111
3990 DATA Ø11ØØ111
4000 DATA Ø11ØØ111
4010 DATA Ø11ØØ111
4020 DATA ØØØØØØØØ
4030 REM N
4040 DATA 78
4050 DATA Ø11ØØ111
4060 DATA Ø111Ø111
4070 DATA Ø1111111
4080 DATA Ø11Ø1111
4090 DATA Ø11ØØ111
4100 DATA Ø11ØØ111
4110 DATA Ø11ØØ111
4120 DATA ØØØØØØØØ
4130 REM O
4140 DATA 79
4150 DATA Ø1111111
4160 DATA Ø11ØØØ11
4170 DATA Ø11ØØØ11
4180 DATA Ø11ØØ111
4190 DATA Ø11ØØ111
4200 DATA Ø11ØØ111
4210 DATA Ø1111111
4220 DATA ØØØØØØØØ
```

```
4230 REM P
4240 DATA 80
4250 DATA Ø1111111
4260 DATA Ø11ØØØ11
4270 DATA Ø11ØØØ11
4280 DATA Ø1111111
4290 DATA Ø111ØØØØ
4300 DATA Ø111ØØØØ
4310 DATA Ø111ØØØØ
4320 DATA ØØØØØØØØ
4330 REM Q
4340 DATA 81
4350 DATA Ø1111111
4360 DATA Ø11ØØØ11
4370 DATA Ø11ØØØ11
4380 DATA Ø11ØØ111
4390 DATA Ø11ØØ111
4400 DATA Ø11ØØ111
4410 DATA Ø1111111
4420 DATA ØØØØØ111
4430 REM R
4440 DATA 82
4450 DATA Ø111111Ø
4460 DATA Ø11ØØ11Ø
4470 DATA Ø11ØØ11Ø
4480 DATA Ø1111111
4490 DATA Ø111Ø111
4500 DATA Ø111Ø111
4510 DATA Ø111Ø111
4520 DATA ØØØØØØØØ
4530 REM S
4540 DATA 83
4550 DATA Ø1111111
4560 DATA Ø11ØØØØØ
4570 DATA Ø1111111
4580 DATA ØØØØØØ11
4590 DATA Ø111ØØ11
4600 DATA Ø111ØØ11
4610 DATA Ø1111111
4620 DATA ØØØØØØØØ
```

```
4630 REM T
4640 DATA 84
4650 DATA Ø1111111
4660 DATA ØØØ111ØØ
4670 DATA ØØØ111ØØ
4680 DATA ØØØ111ØØ
4690 DATA ØØØ111ØØ
4700 DATA ØØØ111ØØ
4710 DATA ØØØ111ØØ
4720 DATA ØØØØØØØØ
4730 REM U
4740 DATA 85
4750 DATA Ø11ØØ111
4760 DATA Ø11ØØ111
4770 DATA Ø11ØØ111
4780 DATA Ø11ØØ111
4790 DATA Ø11ØØ111
4800 DATA Ø11ØØ111
4810 DATA Ø1111111
4820 DATA ØØØØØØØØ
4830 REM V
4840 DATA 86
4850 DATA Ø11ØØ111
4860 DATA Ø11ØØ111
4870 DATA Ø11ØØ111
4880 DATA Ø11ØØ111
4890 DATA Ø11Ø1111
4900 DATA ØØ11111Ø
4910 DATA ØØØ111ØØ
4920 DATA ØØØØØØØØ
4930 REM W
4940 DATA 87
4950 DATA Ø11ØØ111
4960 DATA Ø11ØØ111
4970 DATA Ø11ØØ111
4980 DATA Ø11Ø1111
4990 DATA Ø1111111
5000 DATA Ø1111111
5010 DATA Ø11ØØ111
5020 DATA ØØØØØØØØ
```

```
5030 REM X
5040 DATA 88
5050 DATA 01110011
5060 DATA 01110011
5070 DATA 01110011
5080 DATA 00111110
5090 DATA 01100111
5100 DATA 01100111
5110 DATA 01100111
5120 DATA 00000000
5130 REM Y
5140 DATA 89
5150 DATA 01100111
5160 DATA 01100111
5170 DATA 01100111
5180 DATA 01111111
5190 DATA 00011100
5200 DATA 00011100
5210 DATA 00011100
5220 DATA 00000000
5230 REM Z
5240 DATA 90
5250 DATA 01111111
5260 DATA 01100110
5270 DATA 01101100
5280 DATA 00011000
5290 DATA 00110111
5300 DATA 01100111
5310 DATA 01111111
5320 DATA 00000000
5330 REM a
5340 DATA 97
5350 DATA 00000000
5360 DATA 00000000
5370 DATA 00111110
5380 DATA 00000110
5390 DATA 01111110
5400 DATA 01100110
5410 DATA 01111110
5420 DATA 00000000
```

```
5430 REM b
5440 DATA 98
5450 DATA 00000000
5460 DATA 01110000
5470 DATA 01110000
5480 DATA 01111110
5490 DATA 01110110
5500 DATA 01110110
5510 DATA 01111110
5520 DATA 00000000
5530 REM c
5540 DATA 99
5550 DATA 00000000
5560 DATA 00000000
5570 DATA 01111100
5580 DATA 01110000
5590 DATA 01110000
5600 DATA 01110000
5610 DATA 01111100
5620 DATA 00000000
5630 REM d
5640 DATA 100
5650 DATA 00000000
5660 DATA 00000110
5670 DATA 00000110
5680 DATA 01111110
5690 DATA 01101110
5700 DATA 01101110
5710 DATA 01111110
5720 DATA 00000000
5730 REM e
5740 DATA 101
5750 DATA 00000000
5760 DATA 00000000
5770 DATA 01111110
5780 DATA 01100110
5790 DATA 01111110
5800 DATA 01100000
5810 DATA 01111110
5820 DATA 00000000
```

```
5830 REM f
5840 DATA 102
5850 DATA 00000000
5860 DATA 00011110
5870 DATA 00011000
5880 DATA 00111110
5890 DATA 00011000
5900 DATA 00011000
5910 DATA 00011000
5920 DATA 00000000
5930 REM g
5940 DATA 103
5950 DATA 00000000
5960 DATA 00000000
5970 DATA 01111110
5980 DATA 01101110
5990 DATA 01101110
6000 DATA 01111110
6010 DATA 00001110
6020 DATA 01111110
6030 REM h
6040 DATA 104
6050 DATA 00000000
6060 DATA 01110000
6070 DATA 01110000
6080 DATA 01111100
6090 DATA 01110110
6100 DATA 01110110
6110 DATA 01110110
6120 DATA 00000000
6130 REM i
6140 DATA 105
6150 DATA 00000000
6160 DATA 00011000
6170 DATA 00000000
6180 DATA 00011000
6190 DATA 00011000
6200 DATA 00111100
6210 DATA 00111100
6220 DATA 00000000
```

```
6230 REM j
6240 DATA 106
6250 DATA 00000000
6260 DATA 00001110
6270 DATA 00000000
6280 DATA 00001110
6290 DATA 00001110
6300 DATA 00001110
6310 DATA 00001110
6320 DATA 00111110
6330 REM k
6340 DATA 107
6350 DATA 00000000
6360 DATA 01100000
6370 DATA 01100000
6380 DATA 01101100
6390 DATA 01111000
6400 DATA 01101100
6410 DATA 01100110
6420 DATA 00000000
6430 REM l
6440 DATA 108
6450 DATA 00000000
6460 DATA 00111100
6470 DATA 00011100
6480 DATA 00011100
6490 DATA 00011100
6500 DATA 00011100
6510 DATA 00011100
6520 DATA 00000000
6530 REM m
6540 DATA 109
6550 DATA 00000000
6560 DATA 00000000
6570 DATA 01100111
6580 DATA 01111111
6590 DATA 01111111
6600 DATA 01101011
6610 DATA 01100011
6620 DATA 00000000
```

```
6630 REM n
6640 DATA 110
6650 DATA 00000000
6660 DATA 00000000
6670 DATA 01111110
6680 DATA 01111110
6690 DATA 01100110
6700 DATA 01100110
6710 DATA 01100110
6720 DATA 00000000
6730 REM o
6740 DATA 111
6750 DATA 00000000
6760 DATA 00000000
6770 DATA 01111110
6780 DATA 01101110
6790 DATA 01101110
6800 DATA 01101110
6810 DATA 01111110
6820 DATA 00000000
6830 REM p
6840 DATA 112
6850 DATA 00000000
6860 DATA 00000000
6870 DATA 01111110
6880 DATA 01110110
6890 DATA 01110110
6900 DATA 01111110
6910 DATA 01110000
6920 DATA 01110000
6930 REM q
6940 DATA 113
6950 DATA 00000000
6960 DATA 00000000
6970 DATA 01111110
6980 DATA 01101110
6990 DATA 01101110
7000 DATA 01111110
7010 DATA 00001110
7020 DATA 00001110
```

```
7030 REM r
7040 DATA 114
7050 DATA 00000000
7060 DATA 00000000
7070 DATA 01111110
7080 DATA 01110110
7090 DATA 01110000
7100 DATA 01110000
7110 DATA 01110000
7120 DATA 00000000
7130 REM s
7140 DATA 115
7150 DATA 00000000
7160 DATA 00000000
7170 DATA 01111110
7180 DATA 01100000
7190 DATA 01111110
7200 DATA 00001110
7210 DATA 01111110
7220 DATA 00000000
7230 REM t
7240 DATA 116
7250 DATA 00000000
7260 DATA 00111000
7270 DATA 01111110
7280 DATA 00111000
7290 DATA 00111000
7300 DATA 00111000
7310 DATA 00111110
7320 DATA 00000000
7330 REM u
7340 DATA 117
7350 DATA 00000000
7360 DATA 00000000
7370 DATA 01101110
7380 DATA 01101110
7390 DATA 01101110
7400 DATA 01101110
7410 DATA 01111110
7420 DATA 00000000
```

```
7430 REM v
7440 DATA 118
7450 DATA 00000000
7460 DATA 00000000
7470 DATA 01101110
7480 DATA 01101110
7490 DATA 01101110
7500 DATA 00111100
7510 DATA 00011000
7520 DATA 00000000
7530 REM w
7540 DATA 119
7550 DATA 00000000
7560 DATA 00000000
7570 DATA 01100011
7580 DATA 01101011
7590 DATA 01111111
7600 DATA 01111111
7610 DATA 00110110
7620 DATA 00000000
7630 REM x
7640 DATA 120
7650 DATA 00000000
7660 DATA 00000000
7670 DATA 01100110
7680 DATA 00111100
7690 DATA 00011000
7700 DATA 00111100
7710 DATA 01100110
7720 DATA 00000000
7730 REM y
7740 DATA 121
7750 DATA 00000000
7760 DATA 00000000
7770 DATA 01101110
7780 DATA 01101110
7790 DATA 01101110
7800 DATA 01111110
7810 DATA 00001110
7820 DATA 01111110
```

```
7830 REM z
7840 DATA 122
7850 DATA 00000000
7860 DATA 00000000
7870 DATA 01111110
7880 DATA 00001100
7890 DATA 00011000
7900 DATA 00110000
7910 DATA 01111110
7920 DATA 00000000
7930 a$=INKEY$:IF a$="" THEN GOTO 7930
7940 PRINT ASC(a$):GOTO 7930
```

Errormeldungen

Manchmal ist es wie verhext: Man gibt sein lange Zeit ausgearbeitetes BASIC-Programm in den Computer ein und trotz größter Bemühungen, hier und da treten laufend Fehler auf.

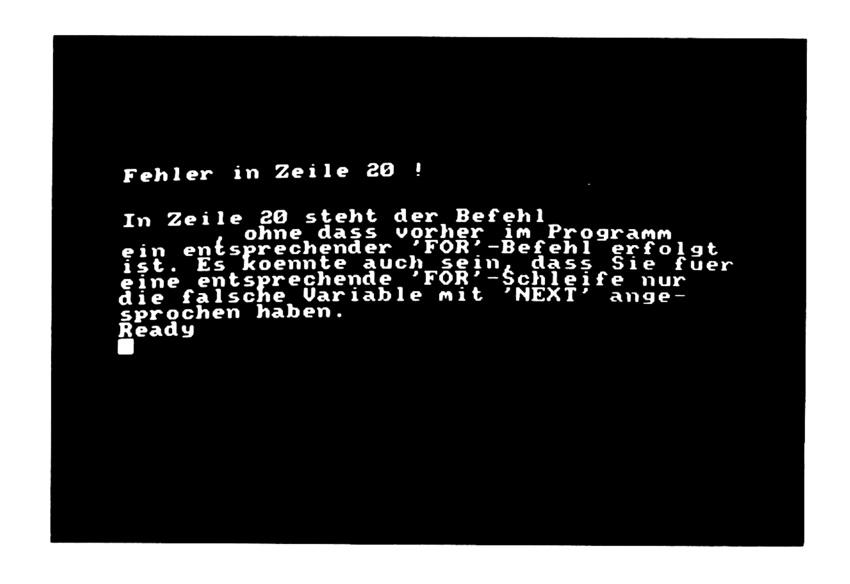
Da der CPC ursprünglich nicht aus Deutschland stammt, werden die Fehlermeldungen selbstverständlich in englischer Sprache ausgeworfen. Da der CPC zudem nicht über unbegrenzten Speicherraum verfügt, sind diese Fehlermeldungen oft nur ein allzu kurzer Hinweis ('Syntax error', 'Type mismatch' ...).

Diesen Hinweis lernt der eifrige Computerprogrammierer mit der Zeit richtig kennen und zu verstehen, aber bis dahin ist es oft noch ein weiter Weg. Aus diesem Grund haben wir ein ausführliches Errormeldungsprogramm in diese Programmsammlung mit aufgenommen. Da der CPC den Vorgabebefehl 'ON ERROR GOTO' kennt, kann er bei einem auftretenden Fehler so zu reagieren lernen, wie wir das wollen (er kann sogar selbst die Fehler teilweise korrigieren; bestes Beispiel dazu: Der CPC lädt Daten von Kassette; schließlich sind alle vorhandenen Daten eingelesen; normalerweise wird nun bei weiteren Leseversuchen die Fehlermeldung ausgegeben: 'EOF met'('EOF' = End of file); viel einfacher ist es da doch, in einer Errorbehandlungsroutine nach Auftreten dieses Fehlers ('ON ERROR GOTO') den noch offenen Datenfile zu schließen und im Programm dort fortzufahren, wo dies im Augenblick sinnvoll erscheint.

Zwar merzt unser Errormeldungsprogramm keine vorhandenen Errors aus (das kann pauschal so gehandhabt leicht zu unvorhersehbaren Folgen führen), dafür hilft es aber bei der Suche nach dem Fehler um so effektiver. Am besten Sie probieren es einmal aus! Geben Sie in Zeile 20 nur 'NEXT n' ein. Starten wir das Programm, so wird gleich der Fehler (Error 1 = 'Unexpected NEXT') angezeigt und eine Lösungsstrategie zur Programmverbesserung angeboten.

Wie verfahren Sie mit Ihren bereits bestehenden Programmen? Zeilen so umnummerieren (Befehl 'RENUM'), daß sich keine Zeile mit dem Errorbehandlungsprogramm überschneidet (die Zeilennummern müssen größer als 10 und kleiner als 10000 sein). Nun gebrauchen Sie den Befehl 'MERGE' (siehe Handbuch), um die beiden Programme miteinander zu verbinden; anschließend mit 'RUN' starten und versuchen, die Fehlermeldungen mit den vorgegebenen Korrekturageboten positiv anzuwenden.

Neben dem BASIC-Befehl 'ON ERROR GOTO' hilft uns der CPC am besten weiter durch die Abfrage der Error-Zeilenvariable 'ERL' und die Fehlermeldevariable 'ERR'. Fehler selbst erzeugen können wir schließlich mit dem Befehl 'ERROR'.



```
10 REM Errormeldungen
11 REM CPC464 Basic Programme
12 REM Copyright 1984 DATA BECKER &
                       Rainer Lueers
13 INK 0,1:INK 1,24:INK 3,1,24
14 ON ERROR GOTO 10000
20 REM Bitte Programm zwischen Zeile
       20 und Zeile 9999 eintippen
       oder einMERGEN
9999 STOP
10000 REM Beginn der ausfuehrlichen
          Errorbehandlungsroutine
10010 MODE 1:PEN 1
10020 PRINT"Fehler in Zeile"ERL"!"
10030 GOSUB 11420
10040 REM Unexpected NEXT
10050 IF ERR=1 THEN PRINT "In Zeile"ERL"
steht der Befehl ":PEN 3 ELSE GOTO 10100
10060 PRINT "'NEXT'";:PEN 1:PRINT ", ohn
e dass vorher im Programm"
10070 PRINT "ein entsprechender 'FOR'-Be
fehl erfolgt ist. Es koennte auch sein,
dass Sie fuereine entsprechende 'FOR'-Sc
hleife nur   die falsche Variable mit 'N
EXT' ange-   sprochen haben."
10080 END
10090 REM Syntax Error
10100 IF ERR=2 THEN PRINT "Irgendetwas i
st":PRINT "im Satzbau von Zeile"ERL ELSE
 GOTO 10150
10110 PRINT "falsch bzw. irgendein Wort
ist falsch   geschrieben. Man spricht vo
n einem      ":PEN 3:PRINT "SYNTAX ERROR"
;:PEN 1:PRINT "."
10120 PRINT
10130 END
10140 REM Unexpected RETURN
```

```
10150 IF ERR=3 THEN PRINT "In Zeile"ERL"
steht der Befehl":PEN 3:PRINT "RETURN";:
PEN 1:PRINT ", ohne dass vorher im" ELSE
 GOTO 10190
10160 PRINT "Programm ein entsprechendes
 Unterpro-   gramm mit 'GOSUB' aufgerufe
n wurde."
10170 END
10180 REM DATA exhausted
10190 IF ERR=4 THEN PRINT "Es sollen nac
h Ihren Wuenschen noch      weitere ";:PE
N 3:PRINT"DATAS";:PEN 1:PRINT " eingeles
en werden." ELSE GOTO 10230
10200 PRINT "Dafuer reicht die angegeben
e Menge der  DATAS aber nicht aus. Viell
eicht wurde  auch zum nochmaligen Lesen
der DATAS    mit dem Befehl 'READ' nur e
in entspre-  chender DATAruecksetzbefehl
 wie         'RESTORE' vergessen."
10210 END
10220 REM Improper argument
10230 IF ERR=5 THEN PRINT "Ein Parameter
 oder ein ";:PEN 3:PRINT "Argument";:PEN
 1:PRINT " wur-" ELSE GOTO 10270
10240 PRINT "de falsch angegeben. Sie ko
ennen        z.B. nicht beliebige Zahlwe
rte fuer den 'SOUND' oder den 'ENV'-Befe
hl verwenden.Bitte schauen Sie im Handbu
ch nach!"
10250 END
10260 REM Overflow
10270 IF ERR=6 THEN PRINT "Der Schneider
-Computer registriert      einen ";:PEN
3:PRINT "OVERFLOW";:PEN 1:PRINT"; das be
deutet einen" ELSE GOTO 10320
```

```
10280 PRINT "Ueberlauf, da in Zeile"ERL"
versucht":PRINT "wurde, eine Zahl darzus
tellen, die       ueber die Darstellungsm
oeglichkeit von  Zahlen hinausging (groe
sser              als 1.7 E-38). Vielleic
ht haben Sie auchdie Umwandlung von Zahl
en in andere"
10290 PRINT "Zahldarstellungen bei unerl
aubter        Groesse angestrebt."
10300 END
10310 REM Memory full
10320 IF ERR=7 THEN PRINT "Speicher oder
 englisch ";:PEN 3:PRINT "MEMORY";:PEN 1
:PRINT " voll." ELSE GOTO 10380
10330 PRINT "Das ist ein starkes Stueck,
 wenn Sie denSpeicher mit einem reinen B
ASIC-Programmvollgeschrieben haben. Am b
esten, Sie   schauen das Programm noch e
inmal durch, wo vielleicht Kuerzungsmoeg
lichkeiten   bestehen z.B. sind 'REM'-Ze
ilen nur zur"
10340 PRINT "Dokumentation des Programms
 noetig.      Uebrigens ... Sie kommen sc
hnell zu ei- nem vollen Speicher, wenn S
ie die Dimen-sionen von Variablen zu hoc
h auslegen   oder nacheinander in zu vie
le Unterpro- gramme oder 'FOR...NEXT'-Sc
hleifen ver-"
10350 PRINT "zweigen."
10360 END
10370 REM Line does not exist
10380 IF ERR=8 THEN PRINT "Sie wollen ei
ne Zeile im Programm an-   springen, die
 es ueberhaupt nicht gibt  oder in engli
sch: ";:PEN 3:PRINT "LINE DOES NOT EXIST
";:PEN 1:PRINT "." ELSE GOTO 10420
10390 PRINT "Vielleicht haben Sie da ein
en Schritt   zu weit gedacht oder gar vo
rhin das Pro-gramm umnummeriert und die
gleiche Feh- lermeldung (s.o.) ignoriert
."
```

```
10400 END
10410 REM Subscript out of range
10420 IF ERR=9 THEN PRINT "Das ";:PEN 3:
PRINT "SUBSCRIPT";:PEN 1:PRINT " (die Gr
oesse der gewaehl-" ELSE GOTO 10470
10430 PRINT "ten Dimension einer Variabl
en) ist in   Zeile";ERL;"zu gross. Entwe
der haben Sie":PRINT "vorhin vergessen,
die Variablen mit     'DIM' zu dimension
ieren oder aber Sie   haben gedacht, der
 Schneider-Computer"
10440 PRINT "koenne mehr als 11 Variable
n, die einer Variablen zugeordnet sind o
hne Dimen-   sionierung verkraften."
10450 END
10460 REM Array already dimensioned
10470 IF ERR=10 THEN PRINT "Die Dimensio
nierung mit ";:PEN 3:PRINT "DIM";:PEN 1:
PRINT " wurde" ELSE GOTO 10510
10480 PRINT "schon einmal vorher im Prog
ramm fest-   gelegt. Es ist nicht moegli
ch, in Zei-  le";ERL;"noch einmal, ohne
vorher":PRINT "'CLEAR' gesagt zu haben,
neu zu dimen-  sionieren."
10490 END
10500 REM Division by zero
10510 IF ERR=11 THEN PEN 3:PRINT "DIVISI
ON";:PEN 1:PRINT " durch 0 ist nicht erl
aubt.":PRINT "Hier ist es dennoch gesche
hen! Bitte im Programm nachsehen, wie es
 dazu kommen  konnte (auch Variablen in
der Fehlerzei-le ueberpruefen)." ELSE GO
TO 10540
10520 END
10530 REM Invalid direct command
10540 IF ERR=12 THEN PRINT "Als direkte
Befehlseingabe (";:PEN 3:PRINT "direct":
PRINT "command";:PEN 1:PRINT ") ist dies
 so nicht statthaft" ELSE GOTO 10580
```

```
10550 PRINT "(z.B. 'A$=INKEY$' zur Tasta
turabfrage). Bitte diesen Befehl in ein
Programm ein-binden und dieses mit 'RUN'
 starten."
10560 END
10570 REM Type mismatch
10580 IF ERR=13 THEN PRINT "Sie haben de
n falschen Variablentyp":PRINT "(";:PEN
3:PRINT "TYPE MISMATCH";:PEN 1:PRINT ")
ausgewaehlt.":PRINT "Entweder fragt man
mit einer Variablen  nur nach einer Zahl
 (Variable A,B,C...)" ELSE GOTO 10630
10590 PRINT "und liest danach auch nur e
ine Zahl ein (durch 'READ' oder 'INPUT')
, oder man   erwartet ein alphanumerisch
es Zeichen   bzw. eine Zahl und liest di
eses mit     einer Stringvariablen (A$,B
$,C$...) ein.";
10600 PRINT "Will man eine Zahl als Stri
ng einlesen, so kann der String durch An
wendung des  Befehls 'VAL(...)' wieder i
n eine Zahl- variable zurueckverwandelt
werden."
10610 END
10620 REM String space full
10630 IF ERR=14 THEN PEN 3:PRINT "STRING
S";:PEN 1:PRINT " finden keinen freien P
latz mehr":PRINT "im Speicher. Es gibt z
wei Moeglichkei-  ten zur weiteren Strin
geingabe:" ELSE GOTO 10670
10640 PRINT:PRINT "1) Das BASIC-Programm
 verkuerzen          2) Die vorhandenen St
rings auf Kassette";:PRINT "    abspeiche
rn und mit 'CLEAR' den          Stringspe
icher loeschen. Danach kann     die Neuei
ngabe der Variablen             erfolgen.
"
10650 END
10660 REM String too long
```

```
10670 IF ERR=15 THEN PRINT "Die in der Z
eile";ERL;"verwendete":PRINT"Zeichenkett
e (";:PEN 3:PRINT "STRING";:PEN 1:PRINT
") ist laenger als" ELSE GOTO 10710
10680 PRINT "255 Zeichen. Dies ist unbed
ingt zu ver- meiden! Deshalb nicht belie
big Strings  erweitern (durch '+') und a
uch nicht zu oft die Stringmanipulations
funktionen   wie 'INSTR' usw. als Vergro
esserungs-   faktor einsetzen."
10690 END
10700 REM String expression too complex
10710 IF ERR=16 THEN PRINT "Die Zeichenk
ette in Zeile";ERL:PRINT "ist ";:PEN 3:P
RINT "zu komplex";:PEN 1:PRINT ". Bitte
nicht zu sehr" ELSE GOTO 10750
10720 PRINT "den String kuenstlich verko
mplizieren   durch Eingaben von Stringma
nipulations- funktionen wie 'LEFT$', 'RI
GHT$', 'MID$'oder 'INSTR'.
10730 END
10740 REM Cannot CONTinue
10750 IF ERR=17 THEN PEN 3:PRINT "CONT";
:PEN 1:PRINT "inue ist zwar eine tolle F
unktion" ELSE GOTO 10800
10760 PRINT "die eine Fortsetzung nach '
END', 'STOP' oder Programmunterbrechung
durch Druek- ken der <ESC>-Taste fast im
mer ermoeg-  licht, aber bitte VORSICHT!
"
10770 PRINT "In diesem Fall ist seit der
 Programmun- terbrechung zu viel gescheh
en, als dass der Computer sich noch dara
n erinnert,  in welcher Zeile er mit dem
 Programmab- lauf fortfahren soll."
10780 END
10790 REM Unknown user function
10800 IF ERR=18 THEN PEN 3:PRINT "FN";:P
EN 1:PRINT " ist zwar ein toller Befehl,
 um mathe-"; ELSE GOTO 10860
```

```
10810 PRINT "matische Funktionen, die du
rch 'DEF FN' an anderer Stelle im Progra
mm definiert wurden, mit beliebigen Vari
ablen auszu- rechnen. Aber in diesem Fal
l wurde in   Zeile";ERL;"mit FN eine Fun
ktion"
10820 PRINT "aufgerufen, die vorher uebe
rhaupt         nicht so definiert worden i
st.":PRINT:PRINT "Ein Hinweis der Uebers
ichtlichkeit      halber: Definitionen s
ollten immer zu   Beginn des Programms i
m sogenannten De-"
10830 PRINT "klarationsteil festgelegt w
erden."
10840 END
10850 REM RESUME missing
10860 IF ERR=19 THEN PEN 3:PRINT "RESUME
";:PEN 1:PRINT " sollte bei einer Interr
uptsteu-" ELSE GOTO 10910
10870 PRINT "erung mit 'ON ERROR GOTO' v
erwendet wer-den, um in den aktuellen Pr
ogrammablauf an der richtigen Stelle wie
der ein-      schwenken zu koennen. Sie k
oennen ein-  fach RESUME im Programm ein
geben, so er-folgt die Programmweiterfue
hrung an der"
10880 PRINT "Stelle, wo der Fehler aufge
treten ist.  Sie koennen auch RESUME Zei
lennummer    eingeben, um an einer gewue
nschten Pro- grammzeile fortzufahren."
10890 END
10900 REM Unexpected RESUME
10910 IF ERR=20 THEN PRINT "Der Befehl "
;:PEN 3:PRINT "RESUME";:PEN 1:PRINT " is
t in Zeile" ELSE GOTO 10960
```

```
10920 PRINT ERL;"aufgetreten, ohne dass
eine Feh-":PRINT "lerroutine (ausgeloest
 durch den Befehl 'ON ERROR GOTO') im Pr
ogramm aufgerufen wurde. Die Loesung des
 anscheinend      ueberfluessigen RESUME
 koennte daraus   resultieren, dass das
eigentliche"
10930 PRINT "Programm nicht mit 'END' ab
geschlossen  wurde und somit nun in die
Fehlerbe-    handlungsroutine reinrutsch
t."
10940 END
10950 REM Direct command
10960 IF ERR=21 THEN PRINT "Waehrend der
 Kassettenladung eines Pro- gramms wurde
 ein Befehl gefunden (";:PEN 3:PRINT "DI
-":PRINT "RECT COMMAND";:PEN 1:PRINT "),
 der ohne Zeilennummer im" ELSE GOTO 110
00
10970 PRINT "Programm auftaucht. Das dar
f nicht so   sein!"
10980 END
10990 REM Operand missing
11000 IF ERR=22 THEN PRINT "Manche BASIC
-Befehle ergeben ohne Ver-  wendung eine
r Mindestanzahl von ";:PEN 3:PRINT "OPE-
":PRINT "RANDEN";:PEN 1:PRINT " keinen S
inn. Z.B. der Befehl" ELSE GOTO 11030
11010 PRINT "'SOUND' muss mindestens aus
 der Angabe  des gewaehlten Tonkanals un
d der Ton-   frequenz bestehen.":PRINT "
Z.B. 'SOUND 1,100' und nicht 'SOUND'."
11020 REM Line too long
11030 IF ERR=23 THEN PRINT "Die Programm
zeile ist zu lang (in eng-  lisch:";:PEN
 3:PRINT "too long";:PEN 1:PRINT "); sie
 darf aus hoech-" ELSE GOTO 11070
11040 PRINT "stens bis zu 255 Zeichen be
stehen.":PRINT:PRINT "Bitte korrigieren
z.B. durch eine Auf-  teilung dieser Zei
```

```
le in mehrere kleine  Zeilen."
11Ø5Ø END
11Ø6Ø REM EOF met
11Ø7Ø IF ERR=24 THEN PEN 3:PRINT "EOF";:
PEN 1:PRINT " bedeutet END OF FILE." ELS
E GOTO 1115Ø
11Ø8Ø PRINT "Es wurde irrtuemlicherweise
 in Zei-      le";ERL;"versucht, eine Dat
ei laenger"
11Ø9Ø PRINT "in den Computer einzuladen,
 als sie     wahrlich lang ist.":PRINT:P
RINT "Um diesen Fehler zu umgehen, gibt
es    folgende Ratschlaege:"
111ØØ PRINT "Vor dem Einlesen von Daten
von der Kas- sette in einer Zeile nachfr
agen:         'IF EOF THEN CLOSE IN:GOTO
...'
1111Ø PRINT "d.h. wenn EOF erreicht wird
, wird die   Datei ordnungsgemaess gesch
lossen und   der Programmablauf an ander
er Stelle    fortgesetzt. Andere Moeglic
hkeit: Die   Anzahl der speichernden Fil
es zuerst in der Datei ablegen. Beim Lad
en zuerst"
1112Ø PRINT "diese Zahl einlesen und die
selbige als  Zaehler fuer eine 'FOR...NE
XT-Schleife  verwenden."
1113Ø END
1114Ø REM File type error
1115Ø IF ERR=25 THEN PRINT "Die Art des
gewaehlten Files (";:PEN 3:PRINT "FILE":
PRINT "TYPE";:PEN 1:PRINT ") ist nicht k
orrekt." ELSE GOTO 112ØØ
1116Ø PRINT "Das Einlesen von Dateien mi
t dem Befehl 'OPEN IN' ist nur bei ASCII
-Dateien      moeglich. 'LOAD' bzw. 'RUN'
 bzw. 'MERGE'hingegen laden nur Programm
e, die mit   'SAVE' direkt abgespeichert
 wurden (hierwurde naemlich nicht Buchst
abe fuer"
```

```
11170 PRINT "Buchstabe sondern BASICwort
(=Token)      fuer BASICwort verschluesse
lt und so-    mit auch verkuerzt abgespei
chert."
11180 END
11190 REM NEXT missing
11200 IF ERR=26 THEN PRINT "Eine 'FOR...
";:PEN 3:PRINT "NEXT";:PEN 1:PRINT "'-Sc
hleife besteht" ELSE GOTO 11250
11210 PRINT "aus 'FOR' und NEXT. Bei der
 Programm-    schleife in Zeile";ERL;"feh
lt"
11220 PRINT "offensichtlich ein NEXT! Bi
tte nach-     schauen und einsetzen."
11230 END
11240 REM File already open
11250 IF ERR=27 THEN PRINT "Eine Datei,
die  mit dem Befehl ";:PEN 3:PRINT "OPEN
";:PEN 1 ELSE GOTO 11300
11260 PRINT "bereits geoeffnet war, darf
 in Zeile":PRINT ERL;"nicht noch einmal
mit OPEN IN"
11270 PRINT "oder OPEN OUT geoeffnet wer
den. Das ist zudem sinnlos. Vielleicht l
iegt der of-  fensichtliche Fehler darin
begruendet,   dass die Datei in einem and
eren Pro-     grammabschnitt oder nach de
m Neustart    des Programms zwar geoeffne
t"
11280 PRINT "(OPEN IN, OPEN OUT), jedoch
 nicht wie-  der geschlossen ('CLOSE IN'
, 'CLOSE     OUT') wurde."
11290 REM Unknown command
11300 IF ERR=28 THEN PRINT "Ein Befehl i
n Programmzeile";ERL ELSE GOTO 11330
```

```
11310 PRINT "ist in dieser Anwendung dem
 Computer    nicht bekannt (";:PEN 3:PRI
NT "UNKNOWN COMMAND";:PEN 1:PRINT ")."
11320 REM WEND missing
11330 IF ERR=29 THEN PRINT "WHILE...";:P
EN 3:PRINT "WEND";:PEN 1:PRINT " bietet
zwar gute Moeglich-" ELSE GOTO 11380
11340 PRINT "keiten zur Erstellung von s
trukturier-  ten BASIC-Programmen. Sie h
aben jedoch  bis zu Zeile";ERL;"vergesse
n, eine"
11350 PRINT "WHILE...WEND-Schleife im Pr
ogramm mit":PEN 3:PRINT "WEND";:PEN 1:PR
INT " abzuschliessen."
11360 END
11370 REM Unexpected WEND
11380 IF ERR=30 THEN PRINT "WHILE...";:P
EN 3:PRINT "WEND";:PEN 1:PRINT " bietet
zwar gute Moeglich-" ELSE END
11390 PRINT "keiten zur Erstellung von s
trukturier-  ten BASIC-Programmen. Sie h
aben jedoch  bis zu Zeile";ERL;"vergesse
n, eine"
11400 PRINT "WHILE...WEND-Schleife im Pr
ogramm vorherueberhaupt mit ";:PEN 3:PRI
NT "WHILE";:PEN 1:PRINT " zu eroeffnen."
11410 END
11420 PEN 3
11430 REM Die Errormeldungen mit ihrer
            englischen Kurzerklaerung
11440 IF ERR=1 THEN PRINT "Unexpected NE
XT";
11450 IF ERR=2 THEN PRINT "Syntax Error"
;
11460 IF ERR=3 THEN PRINT "Unexpected RE
TURN";
11470 IF ERR=4 THEN PRINT "DATA exhauste
d";
```

```
11480 IF ERR=5 THEN PRINT "Improper Argu
ment";
11490 IF ERR=6 THEN PRINT "Overflow";
11500 IF ERR=7 THEN PRINT "Memory full";
11510 IF ERR=8 THEN PRINT "Line does not
 exist";
11520 IF ERR=9 THEN PRINT "Subscript out
 of range";
11530 IF ERR=10 THEN PRINT "Array alread
y dimensioned";
11540 IF ERR=11 THEN PRINT "Division by
zero";
11550 IF ERR=12 THEN PRINT "Invalid dire
ct command"
11560 IF ERR=13 THEN PRINT "Type mismatc
h";
11570 IF ERR=14 THEN PRINT "String space
 full";
11580 IF ERR=15 THEN PRINT "String too l
ong";
11590 IF ERR=16 THEN PRINT "String expre
ssion too complex";
11600 IF ERR=17 THEN PRINT "Cannot CONTi
nue";
11610 IF ERR=18 THEN PRINT "Unknown user
 function";
11620 IF ERR=19 THEN PRINT "RESUME missi
ng";
11630 IF ERR=20 THEN PRINT "Unexpected R
ESUME";
11640 IF ERR=21 THEN PRINT "Direct comma
nd";
11650 IF ERR=22 THEN PRINT "Operand miss
ing";
11660 IF ERR=23 THEN PRINT "Line too lon
g";
11670 IF ERR=24 THEN PRINT "EOF met";
11680 IF ERR=25 THEN PRINT "File type er
ror";
```

```
11690 IF ERR=26 THEN PRINT "NEXT missing
";
11700 IF ERR=27 THEN PRINT "File already
 open";
11710 IF ERR=28 THEN PRINT "Unknown comm
and";
11720 IF ERR=29 THEN PRINT "WEND missing
";
11730 IF ERR=30 THEN PRINT "Unexpected W
END";
11740 PEN 1
11750 PRINT:PRINT
11760 RETURN
```

Variablenreferenzliste

Im Programm 'Speicher 4' haben wir bereits kennengelernt, wie die jeweilige Zeilennummer und Zeilenlänge im Speicher abgelegt wird (Speicherstellen 368 und 369 enthalten die Längenangabe der ersten Zeilennummer, Speicherstelle 370 und 371 teilt uns mit, wie die erste Zeilennummer heißt usw.). Mit Programm 'Speicher 4' haben wir zudem eine einfache Möglichkeit an die Hand bekommen, den Speicher nach einem bestimmten Wort mit bis zu sechs Buchstaben zu durchsuchen. Jedoch so mancher unter Ihnen wird sich fragen: 'was kann ich damit denn sinnvoll anfangen?'. Nun ja, im Endprodukt haben wir dadurch einerseits den Speicher etwas näher kennengelernt, andererseits kann uns Programm 'Speicher 4' aber außerdem behilflich dabei sein, wenn wir gerade nach solch einer alphanumerischen Zeichenfolge suchen.

Im Programm 'Variablenreferenzliste' gehen wir mit ähnlichem Fundament an den Start, aber wieder folgt eine entscheidende Erweiterung: diesmal wird der Speicher nicht nur nach bestimmten Zeichenfolgen durchsucht, nein, es wird zudem eine Ordnung hergestellt.

Die gesuchten Zeichenfolgen setzen sich aus einer ersten Zahl (2, 3, 4 oder 13) und einer zweiten Zahl (0) zusammen. Diese Folge deutet darauf hin, daß hier eine Variable mit entsprechendem Variablennamen im Programm abgelegt wurde.

Da der CPC die Möglichkeit bietet, Variablennamen mit bis zu vierzig Zeichen zu unterscheiden, müssen wir selbstverständlich den ganzen Variablennamen im Speicher aufsuchen. Dabei hilft uns (und natürlich insbesondere sich selber) der CPC ein starkes Stück weiter: während die ersten Zeichen eines Variablennamens immer normal abgespeichert sind, wird beim letzten Zeichen des Variablennamens jeweils das entsprechende ASCII-Zeichen + 128 im Speicher abgelegt. Besteht der Variablenname lediglich aus einem Zeichen (z.B. 'a', 'b' oder 'c'), so gilt selbstverständlich das erste Variablenzeichen gleichzeitig als das letzte, und dies wird dann auch entsprechend abgespeichert (ASCII-Zeichen + 128).

Da es verschiedene Variablentypen gibt (String, ganze Zahl ...), wird dies bei der Abspeicherung mitberücksichtigt und bei unserer Auflistung auch angezeigt.

Was kann unser 'Variablenreferenzlistenprogramm' nun alles von dem und noch darüberhinaus?

1) Es werden Variablen im Programmspeicher gesucht (Start des Programms 'Variablenreferenzliste' mit 'RUN 10000') - bis Zeile 9999 kann Ihr Programm lang sein.

2) Steht der gefundene Variablenname alphabetisch an erster Stelle (von allen in Ihrem Program vorkommenden Variablen), so wird im Speicher = Programm weitergesucht, in welchen Programmzeilen dieser gleiche Name zudem auftaucht. Auch dies wird am Bildschirm angezeigt. So wird mit der Zeit eine alphabetische Variablenliste auf dem Bildschirm erzeugt, die uns bei unserem 'Basic-Programm-Debugging' (Fehlersuche) erheblich weiterhelfen kann.

Probieren Sie das Programm doch gleich aus, indem Sie in Zeile 10 bis 50 folgende Variablenzuweisungen eintragen:

```
'10 a$="CPC464"
 20 b$="Computer"
 30 CPC=464
 40 DATABECKER!=464
 50 BASICPROGRAMME%=464'
```

Nun starten Sie das 'Variablenreferenzlistenprogramm' mit 'RUN 10000' und nach kurzer Zeit werden nicht nur die verwendeten Variablen, sondern auch die ensprechenden Zeilennummern am Bildschirm aufgeführt.

Ist das Programm durchforstet, so schreibt der CPC den verbrauchten Speicherplatz auf den Bildschirm (Angabe für Programm = ohne Referenzlistenprogramm).

Noch ein Hinweis:

Je länger Ihr Programm ist, desto ausführlicher und damit aufwendiger muß der CPC seine Speicherdurchsuchung durchführen. Also: Bitte ein bißchen Geduld.

Die Variablennamen werden erst einmal alphabetisch sortiert (das heißt im Computerhirn nicht nur alphabetisch sondern auch 'vom großen Buchstaben zum kleinen'!), dann ihrer Zeilennummer - bzw. auch mehreren Zeilennummern - zugeordnet und schließlich im Rahmen einer Referenzliste auf den Bildschirm ausgeworfen.

Der angezeigte verbrauchte 'Programmspeicherplatz' bezieht sich nur auf das Programm in den Zeilen bis 10000, denn die 'Referenzliste' gehört ja schließlich nicht zum eigentlichen Hauptprogram dazu; sie fungiert lediglich als Utility!

```
BASICPROGRAMME% 50
CPC 30
DATABECKER! 40
a$ 10
b$ 20

Programmspeicherplatz:      101 Bytes

Ready
list 10-50
10 a$="CPC464"
20 b$="Computer"
30 CPC=464
40 DATABECKER!=464
50 BASICPROGRAMME%=464
Ready
```

```
10000 REM Variablenreferenzliste
10010 REM CPC464 Basic Programme
10020 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
10030 CLEAR:DIM a$(1000)
10040 REM Kleinste und groesste
          moegliche Variable festlegen
10050 merker$=STRING$(40,"z"):merker1$=S
TRING$(40,"!")
10060 MODE 1
10070 z=368:a$="###":b$="#####"
10080 REM Errechnen von Zeilenlaenge
          und Zeilennummer
10090 a=PEEK(z)+256*PEEK(z+1):b=PEEK(z+2
)+256*PEEK(z+3)
10100 REM Berechnung konzentriert sich
      nur auf das Programm oberhalb
      der Variablenreferenzlisten-
      Berechnung
10110 IF b<>10000 THEN GOTO 10150
10120 GOSUB 10360
10130 PRINT:PRINT "Programmspeicherplatz
:  ";USING b$;z-368;:PRINT " Bytes":PRIN
T:END
10140 REM Beginn der Suche nach
          Variablen im Speicher
10150 FOR z1=z+4 TO z+a:IF (PEEK(z1)=13
OR (PEEK(z1)>1 AND PEEK(z1)<5)) AND PEEK
(z1+2)=0 THEN GOSUB 10200
10160 IF z+4<>z+a THEN NEXT z1
10170 z=z+a
10180 GOTO 10090
10190 REM Welche Art von Variable?
          13 = normale Zahl
          3  = String
          2  = ganze Zahl mit '%'
          4  = normale Zahl mit '!'
10200 IF PEEK(z1)=13 THEN zz=13
10210 IF PEEK(z1)=3 THEN zz=3
10220 IF PEEK(z1)=2 THEN zz=2
```

```
10230 IF PEEK(z1)=4 THEN zz=4
10240 REM Laenge des Variablennamens
          untersuchen
10250 IF PEEK(z1+3)>128 THEN zz$=zz$+CHR
$((PEEK(z1+3)-128)) ELSE zz$=zz$+CHR$(PE
EK(z1+3)):z1=z1+1:GOTO 10250
10260 REM Welche Art von Variable?
          13 = normale Zahl
          3  = String
          2  = ganze Zahl mit '%'
          4  = normale Zahl mit '!'
10270 IF zz=13 THEN zz$=zz$
10280 IF zz=3 THEN zz$=zz$+"$"
10290 IF zz=2 THEN zz$=zz$+"%"
10300 IF zz=4 THEN zz$=zz$+"!"
10310 REM Erzeugen einer alphabetischen
          Ordnung, bevor die Variablen-
          namen mit Zeilennummer auf-
          gelistet werden
10320 IF zz$<merker$ AND zz$>merker1$ TH
EN merker$=zz$
10330 IF flag=1 AND zz$=a$(zaehler) THEN
 PRINT b;
10340 zz$="":RETURN
10350 REM Merken des Variablennamens
          und der Zeilennummer, wo er
          auftritt
10360 IF flag=1 THEN flag=0:PRINT:GOTO 1
0070 ELSE flag=1:IF merker$=STRING$(40,"
z") THEN RETURN ELSE PRINT merker$;:zaeh
ler=zaehler+1:a$(zaehler)=merker$:merker
1$=merker$:merker$=STRING$(40,"z")
10370 GOTO 10070
```

Kalender

Wer sich an den Umgang mit Computern gewöhnt hat, wird dem Programm 'Kalender' in dieser oder einer ähnlichen Form häufiger begegnen. Sei es nun das Programm 'Kalender' allein wie in diesem Fall, oder aber als Einbimdung in ein größeres Programm (z.B. 'Terminverwaltung' - auch als Listing in diesem Buch enthalten).

In jedem Fall geht es um die Aufaddition von Jahren, Monaten und Tagen, wobei als Endresultat der gewünschte Wochentag errechnet wird.

Das vorliegende Programm teilt sich in drei Bereiche, die aus der INPUT-Abfrage zum gewnschten Jahr, Monat und Kalendertag resultieren.

Die Jahreseingabe darf zwischen 1901 und dem Jahr 2000 liegen, wobei Schaltjahre automatisch registriert werden (Zeile 140).

Bei der Monatsrechnung wird der eingegebenen Monatszahl entsprechend der Name zugeordnet und die Höchsttageszahl festgestellt. Liegt ein Merker von der Jahreseingabe in der Form z=1 = Schaltjahr und m=2 = Februar vor, so wird der 29.Februar = Schaltjahr mit hinzugezählt (Zeile 590).

Schließlich erfolgt die Verzweigung zu den einzelnen Wochentagen (Zeile 750) und deren entsprechende Anzeige ('Der ... im Jahr ... ist/war ein ...'), und falls erwünscht, so kann wieder von vorne begonnen werden, sprich ein weiterer Wochentag ausgerechnet werden.

```
         Bestimmung des Wochentags
            von 1901 bis 2000


Jahr ? 1984
Monat ? 11
Tag ? 18
         Der 18 .November. im Jahr 1984
             ist/war ein Sonntag


         Wollen Sie noch einen
         Wochentag ausrechnen ( /N) ? █
```

```
10 REM Kalender
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
50 MODE 1
60 PRINT
70 PRINT
80 PRINT
90 PRINT "      Bestimmung des Wochentag
s                      von 1901 bis 2000"
100 PRINT:PRINT
110 REM Eingabe der Jahreszahl und
        Berechnung (u.a. Schaltjahr)
120 INPUT "Jahr ";j
130 IF j<1901 OR j>2000 THEN GOSUB 990:G
OTO 50
140 IF INT(j/4)=j/4 THEN z=1 ELSE z=0
150 j1=2
160 IF j=1901 THEN GOTO 230
170 FOR n=1902 TO j
180 j1=j1+1
190 IF INT((n-1)/4)=(n-1)/4 THEN j1=j1+1
200 IF j1>6 THEN j1=j1-7
210 NEXT n
220 REM Eingabe des Monats und
        Berechnung (Name und Anzahl der
        Tage)
230 INPUT "Monat ";m
240 IF m<1 OR m>12 THEN GOSUB 990:GOTO 5
0
250 ON m GOSUB 270,290,310,330,350,370,3
90,410,430,450,470,490
260 GOTO 510
270 m$="Januar"
280 RETURN
290 m$="Februar"
300 RETURN
310 m$="Maerz"
```

```
320 RETURN
330 m$="April"
340 RETURN
350 m$="Mai"
360 RETURN
370 m$="Juni"
380 RETURN
390 m$="Juli"
400 RETURN
410 m$="August"
420 RETURN
430 m$="September"
440 RETURN
450 m$="Oktober"
460 RETURN
470 m$="November"
480 RETURN
490 m$="Dezember"
500 RETURN
510 RESTORE:FOR n=1 TO m
520 READ m1
530 NEXT n
540 DATA 31,28,31,30,31,30
550 DATA 31,31,30,31,30,31
560 IF z=1 AND m=2 THEN m1=29
570 REM Eingabe des Tages (Zahl) und
        Berechnung des Wochentags
580 INPUT "Tag ";t
590 IF t=29 AND m=2 AND z<>1 THEN GOSUB
990:GOTO 50
600 IF t<1 OR t>m1 THEN GOSUB 990:GOTO 5
0
610 t1=0
620 RESTORE
630 IF m=1 THEN GOTO 690
640 FOR n=2 TO m
650 READ m1
660 t1=t1+m1
670 NEXT n
680 IF z=1 AND m>2 THEN t1=t1+1
```

```
690 t1=t1+t+j1
700 t1=t1-INT(t1/7)*7
710 IF t1=0 THEN t1=7
720 PRINT
730 PRINT "      Der";t;".";m$;". im Jahr
";j
740 PRINT "          ist/war ein ";
750 ON t1 GOSUB 840,860,880,900,920,940,
960
760 PRINT
770 PRINT
780 INPUT "      Wollen Sie noch einen
                 Wochentag ausrechnen ( /
N) ";f$
790 f$=UPPER$(f$)
800 IF f$="N" THEN END
810 RESTORE
820 CLS
830 GOTO 60
840 PRINT "Sonntag"
850 RETURN
860 PRINT "Montag"
870 RETURN
880 PRINT "Dienstag"
890 RETURN
900 PRINT "Mittwoch"
910 RETURN
920 PRINT "Donnerstag"
930 RETURN
940 PRINT "Freitag"
950 RETURN
960 PRINT "Sonnabend"
970 RETURN
980 END
990 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
1000 PEN normal:GOSUB 1010:RETURN
1010 PRINT:PRINT TAB(7) "<Bitte eine Tas
te druecken>"
1020 f$=INKEY$:IF f$="" THEN GOTO 1020
1030 RETURN
```

Kartei für Schallplatten

Dieses Programm trägt seinen Namen nur dann zu recht, wenn Sie es Wort für Wort bzw. Befehl für Befehl so eintippen, wie Sie es in unserer Programmsammlung vorfinden.

Da es sich aber im Endprodukt um eine gar nicht so unluxuriöse kleine Datenbank handelt, ist es auch möglich, die entsprechenden PRINT-Befehle z.B. 'PRINT "Platten eingeben"' in gewünschte Befehle mit nachfolgendem Text umzuwandeln z.B. 'PRINT "Adressen eingeben"'.

So hätten wir nach ein paar Abänderungen im Programm eine Adreßdatei, die wir natürlich auch später so abspeichern können bzw. so abspeichern sollten.

Jedoch zeigt das Programm 'Kartei für Schallplatten' einige Beschränkungen, die sich erst dann sinnvoll beseitigen ließen, wenn wir nicht mit Kassette sondern mit Diskette - und dort mit wahlfreiem Zugriff - arbeiten könnten.

Die Grenzen des Programms zeigen sich nämlich deutlich bei dem zur freien Verfügung stehenden Speicherplatz. Wollen wir eine Karteikarte vollständig nutzen (drei mal 40 Zeichen), so werden dazu bereits 120 Speicherplätze benötigt. Bei 100 Karteikarten sind dies schon 12000 Speicherplätze ... und daraus resultiert, daß wir als Obergrenze vorerst die Zahl 300 für die Menge der unterschiedlichen Karteikarten vorgegben haben.

Speichern Sie pro Karteikarte (120 Speicherplätze) nicht so viele Zeichen ab, können Sie selbstverständlich hierzu auch das Programm entsprechend ändern. In diesem Fall wird die Höchstzahlgrenze einfach größer umschrieben z.B. Zeile 90:

'IF z<1 OR z>400 ...'

Im einzelnen stellt Ihnen das Programm 'Kartei für Schallplatten' folgende Elemente zur Verfügung:

1. Sie können Karteikarten mit je drei Feldern anlegen (Plattenname, Sänger, Nummer)

2. Sie können Dateien von Kassette laden

3. Sie können Dateien auf Kassette abspeichern

4. Sie können nach einer Platte im Speicher suchen lassen (nach Plattenname, Sänger oder Nummer); bei dieser Funktion wird Ihnen der Bildschirm mit gefundenen Daten aufgefüllt, bis er voll ist. Danach drücken Sie die (ENTER)-Taste und die Auflistung geht weiter. Mehrmaliges Auffinden kann möglich sein, wenn Sie mehrere Langspielplatten eines Sängers, z.B. von David Bowie, besitzen.

5. Sie können Ihre gesamte Plattenliste auf den Bildschirm - oder falls vorhanden auf einen angeschlossenen Drucker - ausgeben lassen (bei der Bildschirmausgabe immer nur fünf Datensätze, dann (ENTER) drücken).

6. Sie können eingegebene Daten ändern. Da bereits bei der Eigabe (1.) die Möglichkeit der nachträglichen Korrektur eingeräumt wird ('Richtig?'), wird Funktion 6. wohl kaum angewendet werden müssen, um Eingabefehler nachträglich zu beseitigen. Vielmehr ist bei dieser Funktion daran gedacht, daß Sie das Dateiprogramm z.B. - wie oben bereits vorgeschlagen - zur Verwaltung von Adressen verwenden. Wie oft kommt es da doch vor, daß jemand seinen Wohnsitz gewechselt hat ... Ganz einfach: Adreßdatei aufrufen, Punkt 6 = 'Änderungen vornehmen' aufrufen. Da sich sicherlich nicht alle Daten Ihres Bekannten von heute auf morgen verändert haben (zumindest sein Name nicht), können Sie diese Programmteile durch Drücken der (ENTER)-Taste einfach überspringen. Erst an der Stelle, wo wahrlich eine Änderung aufgetreten ist, können Sie eingeben.

Ob Sie dieses Programm nun wie ursprünglich gedacht für die Speicherung Ihres Schallplattenarchivs verwenden oder ob Sie das Programm in eine Adreßverwaltung ummodeln, auf jeden Fall wünschen wir Ihnen viel Freude dabei, der Unordung der Ordnung halber mit Ihrem CPC und unserem Programm zu Leibe zu rücken.

```
Kartei fuer Schallplatten
          Menue

   1.Platten eingeben
   2.Platten laden
   3.Platten speichern
   4.Nach Platten suchen
   5.Platten ausgeben
   6.Platten aendern

Ihre Wahl (1/2/3/4/5/6) ?
```

```
4.Nach Platten suchen

Zum Menue zurueck durch <ENTER>
ohne Worteingabe

Wonach wollen Sie suchen

1.Plattenname
2.Saenger/in, Gruppe
3.Nummer/Code

Ihre Wahl (1/2/3) ? 1
```

```
10 REM Kartei fuer Schallplatten
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
50 m=1:MODE 1
60 REM Vorbedingungen treffen zum
       Anlegen von dimensionierten
       Variablen
70 PAPER 1:PEN 0:PRINT "Kartei fuer Scha
llplatten":PAPER 0:PEN 1
80 PRINT
90 INPUT "Wie viele LPs sollen denn gesp
eichert   werden (hoechstens 300) ";z
100 IF z<1 OR z>300 THEN GOSUB 2300:GOTO
 50
110 fr=FRE(0):DIM p$(z),s$(z),n$(z)
120 PRINT:PRINT "Allein die Dimensionier
ung von";z;:PRINT "Platten braucht";fr-F
RE(0);"Speicherplaetze."
130 PRINT:PRINT:PRINT "Bleiben pro Platt
e im Hoechstfall":PRINT:PRINT "
  ";INT(FRE(0)/z):PRINT:PRINT "Speicherp
laetze frei.":PRINT:PRINT:PRINT
140 INPUT "Reicht das (sonst muessen Sie
 die        Dimensionierung = Anzahl einz
ugebender  Platten aendern ( /N) ";f$
150 f$=LEFT$(UPPER$(f$),1)
160 IF f$="N" THEN CLEAR:GOTO 50
170 REM Anzeige des Hauptmenues
180 CLS
190 LOCATE 6,8
200 PAPER 1:PEN 0:PRINT "Kartei fuer Sch
allplatten":PAPER 0:PEN 1
210 LOCATE 16,10
220 PRINT "Menue"
230 LOCATE 8,13
240 PRINT "1.Platten eingeben"
```

```
250 LOCATE 8,14:PRINT "2.Platten laden"
260 LOCATE 8,15:PRINT "3.Platten speiche
rn"
270 LOCATE 8,16:PRINT "4.Nach Platten su
chen"
280 LOCATE 8,17:PRINT "5.Platten ausgebe
n"
290 LOCATE 8,18:PRINT "6.Platten aendern
"
300 LOCATE 6,21
310 INPUT "Ihre Wahl (1/2/3/4/5/6) ";f$
320 IF f$<"1" OR f$>"6" THEN GOSUB 2300:
GOTO 180
330 ON VAL(f$) GOSUB 370,680,940,1220,17
50,2020
340 f$="":f1$="":f2$="":n1=0
350 GOTO 180
360 REM Unterprogramm zur Eingabe
        von Daten
370 FOR n=m TO z
380 CLS
390 PAPER 1:PEN 0:PRINT "1.Platten einge
ben":PAPER 0:PEN 1
400 PRINT
410 GOSUB 2360
420 PRINT
430 f1$="":PRINT "Plattenname"
440 INPUT p$(n):IF LEN(p$(n))>40 THEN p$
(n)=LEFT$(p$(n),40)
450 IF p$(n)="" THEN m=n:RETURN
460 PRINT "Saenger/in, Gruppe"
470 INPUT s$(n):IF LEN(s$(n))>40 THEN s$
(n)=LEFT$(s$(n),40)
480 PRINT "Nummer/Code"
490 INPUT n$(n):IF LEN(n$(n))>40 THEN n$
(n)=LEFT$(n$(n),40)
500 REM Erste Anzeige der eingegebenen
        Daten mit
        Korrekturmoeglichkeit
510 PRINT
```

```
520 PRINT "Noch einmal die eingegebenen
Daten:"
530 PRINT "Satznummer";n;"von bis zu";z
540 PRINT
550 PAPER 1:PEN Ø:PRINT "Plattenname:":P
APER Ø:PEN 1
560 PRINT p$(n)
570 PAPER 1:PEN Ø:PRINT "Saenger/in, Gru
ppe:":PAPER Ø:PEN 1
580 PRINT s$(n)
590 PAPER 1:PEN Ø:PRINT "Nummer/Code:":P
APER Ø:PEN 1
600 PRINT n$(n)
610 PRINT
620 INPUT "Richtig ( /N)";f1$
630 f1$=UPPER$(f1$)
640 IF f1$="N" THEN PRINT:GOTO 430
650 NEXT n
660 RETURN
670 REM Unterprogramm zum Laden
        von Dateien von Kassette
680 CLS
690 PAPER 1:PEN Ø:PRINT "2.Platten laden
":PAPER Ø:PEN 1
700 PRINT
710 GOSUB 2360
720 PRINT
730 PRINT "Sie loeschen durch die Dateie
ingabe     alle bereits vorhandenen Date
n          im Speicher!"
740 PRINT
750 INPUT "Wollen Sie Daten laden (J/ )
";f1$
760 f1$=LEFT$(UPPER$(f1$),1)
770 IF f1$="J" THEN GOTO 780 ELSE RETURN
780 PRINT
790 PRINT "Dateiname"
800 INPUT dn$
810 dn$="!"+dn$
820 OPENIN dn$
```

```
830 INPUT #9,m
840 PRINT
850 PRINT "Die einzuladende Datei ";dn$
860 PRINT "besteht aus";m-1;"Saetzen"
870 FOR n=1 TO m-1
880 INPUT #9,p$(n),s$(n),n$(n)
890 PRINT n:PRINT p$(n):PRINT s$(n):PRIN
T n$(n)
900 NEXT n
910 CLOSEIN
920 RETURN
930 REM Unterprogramm zum Speichern
        von Dateien auf Kassette
940 CLS
950 PAPER 1:PEN 0:PRINT "3.Platten speic
hern":PAPER 0:PEN 1
960 PRINT
970 GOSUB 2360
980 PRINT
990 INPUT "Wollen Sie Daten speichern (J
/ ) ";f1$
1000 f1$=UPPER$(f1$)
1010 IF f1$="J" THEN GOTO 1020 ELSE RETU
RN
1020 PRINT
1030 IF dn$<>"" THEN PRINT "Soll unter d
em Dateinamen ";dn$:INPUT "abgespeichert
 werden (J/ ) ";f2$:f2$=LEFT$(UPPER$(f2$
),1):IF f2$="J" THEN GOTO 1070
1040 PRINT "Dateiname ";
1050 INPUT dn$
1060 dn$="!"+dn$
1070 OPENOUT dn$
1080 PRINT #9,m
1090 PRINT
1100 PRINT "Die zu speichernde Datei ";d
n$;:PRINT "besteht aus";m-1;"Saetzen"
1110 FOR n=1 TO m-1
1120 PRINT #9,p$(n),s$(n),n$(n)
1130 PRINT n
```

```
1140 PRINT p$(n)
1150 PRINT s$(n)
1160 PRINT n$(n)
1170 PRINT
1180 NEXT n
1190 CLOSEOUT
1200 RETURN
1210 REM Unterprogramm zum Suchen nach
         Platten, Saengern oder
         Nummern
1220 CLS
1230 PAPER 1:PEN 0:PRINT "4.Nach Platten
 suchen":PAPER 0:PEN 1
1240 PRINT
1250 GOSUB 2360
1260 PRINT
1270 REM Vorgabe eines Untermenues:
         Suche nach:
         1) Plattenname
         2) Saenger/in, Gruppe
         3) Nummer/Code
1280 PRINT "Wonach wollen Sie suchen"
1290 PRINT
1300 PRINT "1.Plattenname"
1310 PRINT "2.Saenger/in, Gruppe"
1320 PRINT "3.Nummer/Code"
1330 PRINT
1340 INPUT "Ihre Wahl (1/2/3) ";f1$
1350 IF f1$<"1" OR f1$>"3" THEN RETURN
1360 ON VAL(f1$) GOSUB 1390,1510,1620
1370 f1$="":f2$="":n1=0:GOTO 1220
1380 REM Suche nach Plattenname
1390 CLS
1400 PRINT "Plattenname"
1410 INPUT p$:IF LEN(p$)>40 THEN p$=LEFT
$(p$,40)
1420 IF p$="" THEN RETURN
1430 FOR n=1 TO m-1
1440 IF p$=p$(n) THEN PRINT n:PRINT p$(n
):PRINT s$(n):PRINT n$(n):n1=n1+1
```

```
1450 IF n1=5 THEN n1=0:GOSUB 2330
1460 NEXT n
1470 PRINT
1480 GOSUB 2330
1490 RETURN
1500 REM Suche nach Saenger/in, Gruppe
1510 CLS
1520 PRINT "Saenger/in, Gruppe"
1530 INPUT s$:IF LEN(s$)>40 THEN s$=LEFT
$(s$,40)
1540 IF s$="" THEN RETURN
1550 FOR n=1 TO m-1
1560 IF s$=s$(n) THEN PRINT n:PRINT p$(n
):PRINT s$(n):PRINT n$(n):n1=n1+1
1570 IF n1=5 THEN n1=0:GOSUB 2330
1580 NEXT n
1590 PRINT
1600 GOSUB 2330
1610 RETURN
1620 CLS
1630 REM Suche nach Nummer/Code
1640 PRINT "Nummer/Code"
1650 INPUT n$
1660 IF n$="" THEN RETURN
1670 FOR n=1 TO m-1
1680 IF n$=n$(n) THEN PRINT n:PRINT p$(n
):PRINT s$(n):PRINT n$(n):n1=n1+1
1690 IF n1=5 THEN n1=0:GOSUB 2330
1700 NEXT n
1710 PRINT
1720 GOSUB 2330
1730 RETURN
1740 REM Unterprogramm zur Ausgabe der
         gespeicherten Daten auf
         Bildschirm oder Drucker
1750 CLS
1760 PAPER 1:PEN 0:PRINT "5.Platten ausg
eben":PAPER 0:PEN 1
1770 PRINT
```

```
1780 GOSUB 2360
1790 PRINT
1800 PRINT "Auf Bildschirm oder Drucker
ausgeben?"
1810 INPUT "(B/D) ";f2$
1820 IF f2$="" THEN RETURN
1830 f2$=LEFT$(UPPER$(f2$),1)
1840 IF f2$="D" THEN GOTO 1950
1850 REM Ausgabe der Daten auf Monitor
1860 CLS:FOR n=1 TO m-1
1870 n1=n1+1
1880 IF n1=5 THEN n1=0:GOSUB 2330:CLS
1890 PRINT n:PRINT p$(n):PRINT s$(n):PRI
NT n$(n)
1900 NEXT n
1910 PRINT
1920 GOSUB 2330
1930 RETURN
1940 REM Ausgabe der Daten auf Drucker
1950 FOR n=1 TO m-1
1960 PRINT #8,n:PRINT #8,p$(n):PRINT #8,
s$(n):PRINT #8,n$(n):PRINT #8
1970 NEXT n
1980 PRINT
1990 GOSUB 2330
2000 RETURN
2010 REM Unterprogramm zur Aenderung
         von Daten im Speicher
2020 CLS
2030 PAPER 1:PEN 0:PRINT "6.Platten aend
ern":PAPER 0:PEN 1
2040 PRINT
2050 GOSUB 2360
2060 PRINT
2070 INPUT "Welche Plattennummer ";n1
2080 IF n1>m-1 OR n1<0 THEN GOSUB 2300:G
OTO 2070
2090 IF n1=0 THEN RETURN
2100 PRINT
```

```
2110 PRINT n1:PRINT p$(n1):PRINT s$(n1):
PRINT n$(n1)
2120 PRINT
2130 PRINT "Was hat sich geaendert (Nein
=<ENTER>)?"
2140 PRINT p$(n1)
2150 INPUT d$:IF d$<>"" THEN p$(n1)=d$
2160 PRINT s$(n1)
2170 INPUT d$:IF d$<>"" THEN s$(n1)=d$
2180 PRINT n$(n1)
2190 INPUT d$:IF d$<>"" THEN n$(n1)=d$
2200 CLS
2210 PRINT "Die Eingabe zu Plattennummer
";n1;":"
2220 PRINT p$(n1)
2230 PRINT s$(n1)
2240 PRINT n$(n1)
2250 PRINT
2260 GOSUB 2330
2270 n1=0
2280 GOTO 2020
2290 REM Unterprogramm zur Anzeige bei
         falscher Eingabe
2300 PRINT:PEN effekt:PRINT TAB(12) "Fal
sche Eingabe!":PEN normal:GOSUB 2330:RET
URN
2310 a$=INKEY$:IF a$="" THEN GOTO 2310 E
LSE RETURN
2320 REM Unterprogramm zum Einfrieren
         der Bildschirmanzeige
2330 PRINT:PRINT TAB(7) "<Bitte eine Tas
te druecken>"
2340 a$=INKEY$:IF a$="" THEN GOTO 2340 E
LSE RETURN
2350 REM Allgemeines Unterprogramm
2360 PRINT "Zum Menue zurueck durch <ENT
ER>           ohne Worteingabe"
2370 RETURN
```

Sporttabelle

Dieses Programm kann Ihnen in Zukunft sehr viel Freude bereiten, und noch mehr Zeit ersparen.

Hier wird genau das vom Computer verwaltet, was sich innerhalb eines Jahres in Sportvereinen so oft und so regelmäßig abspielt. Als Paradebeispiel sei der Fußball genannt, der auch als Vorbild zu diesem Programm diente.

Zu Beginn der Eingabe haben Sie die Wahl, bis zu zwanzig Vereine gleichzeitig zu verwalten. Anschließend fragt Sie der CPC, ob die Vereinsnamen, die in DATA-Zeilen bereits gespeichert sind (1. und 2. Bundesliga), für die Saison zutreffen oder ob Sie selbst die Namen Ihrer bis zu zwanzig Vereine eingeben wollen.

Auch für die Begegnungen der Vereine in der Hin- und Rückrunde sind bereits alle möglichen Paarungen gespeichert. Wollen Sie diese Daten vom Computerspeicher übernehmen, so beantworten Sie auch diese Frage mit j(a).

Hierzu ein Hinweis: Die in DATA-Zeilen gespeicherten Begegnungen der Clubs resultieren aus einer allgemeinen Aufstellung, wobei schließlich jede Mannschaft gegen jede Mannschaft nach einer Saison zweimal gespielt hat, einmal zu Hause, ein anderes Mal auswärts. Wenn Sie die Daten aus den DATA-Zeilen nicht übernehmen wollen, so geben Sie bitte, der Bildschirmanzeige entsprechend, alle Spieltage nacheinander in den Computerspeicher ein. Jedoch brauchen Sie nur die Hälfte aller Spielpaarungen einzugeben, denn die Rückrunde errechnet der CPC selbstverständlich ohne Ihr Zutun.

Anschließend werden Sie mit einem Menü konfrontiert, das Ihnen drei Möglichkeiten eröffnet: 1. Spieltag eingeben 2. aktuelle Tabelle zeigen 3. Spieltag ansehen.

Geben wir also zuerst die Ergebnisse des 1. Spieltags ein. Da es vorkommen kann, daß das eine oder andere Spiel ausfällt, gibt es auch hierfür eine Hilfe: Das Ergebnis '99,99' bedeutet: Spiel ausgefallen.

Wenn nicht bereits der zweite Spieltag erfolgt ist und diese Eintragungen Vorrang genießen, könen wir uns nun unsere Eingaben in übersichtlicher Form auf den Monitor holen. Wie funktioniert das nun: 3. Spieltag ansehen? Uns wird nach der Auswahl dieses Menüpunktes die Möglichkeit eröffnet, nacheinander die ganze Saison auf Tastendruck vor unseren Augen abspielen zu lassen. Wir können aber selbstverständlich auch gezielt einige Spieltage zur Anzeige bringen.

Springen wir in die Rückrunde (bei 20 Vereinen ab Spiel 20), so wird uns das - falls vorhanden - Spiel von der Hinrunde ebenfalls auf dem Bildschirm angezeigt.

Ist eine Begegnung vor ein paar Spieltagen z.B. durch schlechtes Wetter ausgefallen (Eingabe '99, 99'), so können wir die Eintragung dieses Ergebnises selbstverständlich nachholen. Hierzu bitte wieder 1. 'Spieltag eingeben' anwählen und den entsprechenden Spieltag aufrufen, wo die eine oder andere Partie noch aussteht. Der CPC wird mit einem kurzen Piepsen die bereits fertigen Spiele übergehen und genau da anhalten, wo noch Eintragungen vorzunehmen sind! Waren gleich mehrere Partien an einem Spieltag ausgefallen, so müssen die noch nicht nachgespielten Begegnungen auch bei dieser Wiederholungseingabe erneut mit '99, 99' (ausgefallen) quittiert werden.

Nun aber zum Kernstück des Programms, hin zur Tabellenerrechnung. Es dauert ein klein wenig, bis die Tabelle steht, dafür ist sie aber nicht nur geordnet, sondern auch noch sehr ausführlich auf 80 Zeichen dargestellt: Hinter der erreichten Platzziffer und dem Vereinsnamen wird die Anzahl der bereits erfolgten Spiele des jeweiligen Vereins angezeigt; kurz dahinter ist aufgeführt, wie viele Spiele von diesem Verein gewonnen wurden, wie viele verloren gingen und selbstverständlich auch, wie oft unentschieden gespielt wurde! Schließlich wird auch noch das aktuelle Tor- und Punkteverhältnis zur Anzeige gebracht,

jeweils ergänzt durch die positive oder negative Differenz aus Plus- und Minuspunkten.

So bleibt die Übersicht jederzeit gewahr, wie und wo mit welchen Ergebnissen wann gespielt wurde und wie nun die aktuelle Platzziffer der jeweiligen Vereine ist.

Ein aufwendiges Programm, das allerdings eines erfordert: Lassen Sie den CPC unter Strom! Eine Datenspeicherung ist absichtlich noch nicht eingefügt, denn mit Kassette würde dies bei solch einer Datenfülle einfach zu lange dauern. Warten wir geduldig auf die Diskettenstation! Die Anpassung dürfte Ihnen nicht allzu schwerfallen, da wir zu Beginn die dimensionierten Variablen in einer REM-Zeile erklärt haben. Hingegen können Sie einen Printer-Ausdruck der Tabelle ohne weiteres vornehmen, wenn Sie die dort stehenden PRINT-Befehle (Zeile 900 bis 960) um '#8' ergänzen.



Spieltag 20

1	Duisburg	Hannover	1	:	1	7	:	1
2	St.Pauli	Aachen	2	:	2	3	:	3
3	BW 90	Hertha	3	:	3	4	:	5
4	Stuttga.	Wattens.	4	:	4	-	:	-
5	Oberhau.	Saarbru.	3	:	3	-	:	-
6	Darmsta.	Solingen	6	:	2	3	:	4
7	Freiburg	Kassel	3	:	5	5	:	4
8	Koeln	Nuernbe.	4	:	6	6	:	6
9	Ulm	Homburg	2	:	2	2	:	3
10	Offenba.	Buersta.	3	:	1	3	:	4

<Bitte eine Taste druecken>

```
10 REM Sporttabelle
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 INK Ø,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1:PEN 1:PAPER Ø
50 MODE 1
60 REM Auswahl zwischen 4 bis
       20 Vereinen
70 INPUT "Vereine (4/6/8/10/12/14/16/18/
20) ";f$
80 REM Liste der Variablen, die
       dimensioniert werden mit
       ihren Bedeutungen
90 REM a$( )=Vereinsnamen
       b( , )=Spiel stattgefunden
       c( , , )=Ergebnis
       e( , )=Tore insg.
       f( , )=reingelassen insg.
100 REM g( , )=gewonnen insg.
        h( , )=unentschieden insg.
        i( , )=verloren insg.
        k( , , )=Spielpaarung
        z1 bis z5=fuer Sortierroutine
110 f=VAL(f$):IF f/2<>INT(f/2) THEN PRIN
T:GOSUB 1290:GOTO 50
120 IF f<4 OR f>20 THEN GOSUB 1290:GOTO
50
130 DIM a$(f),b(f*2,f/2),c(f,f*2-2,2),e(
f),f(f),g(f),h(f),i(f),k(f-1,f/2,2),z1(f
+1),z2(f+1),z3(f+1),z4(f+1),z5(f+1)
140 REM Eingabe eigener Vereinsnamen
        oder einlesen der Vereinsnamen
        aus DATA-Zeilen
150 INPUT "Namen aus Programmzeilen (J/
)";f$
160 f$=UPPER$(f$):IF LEFT$(f$,1)="J" THE
N GOTO 230
170 FOR n=1 TO f
180 PRINT "Verein";n;:INPUT ": ";f$
```

```
190 IF f$="" THEN GOTO 180
200 IF LEN(f$)>10 THEN a$(n)=LEFT$(f$,10
) ELSE a$(n)=f$
210 NEXT n
220 GOTO 350
230 ON f GOSUB 250,250,250,250,250,260,2
50,270,250,280,250,290,250,300,250,310,2
50,320,250,330
240 GOTO 350
250 RESTORE 6350:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
260 RESTORE 6320:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
270 RESTORE 6290:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
280 RESTORE 6260:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
290 RESTORE 6230:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
300 RESTORE 6200:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
310 RESTORE 6170:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
320 RESTORE 6130:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
330 RESTORE 6080:FOR n=1 TO f:READ a$(n)
:NEXT n:RETURN
340 REM Spielfolge in DATA-Zeilen
        (uebliches System) kann heran-
        gezogen werden; es ist aber
        auch moeglich, alle Spielzu-
        sammenstellungen selbst
        einzugeben
350 INPUT "Spielfolge wie in DATA-Zeilen
 (J/ )";f$:f$=UPPER$(f$):IF LEFT$(f$,1)=
"J" THEN GOTO 420
360 PRINT:PRINT:PRINT "Spielfolge bitte
mit Zahlen eingeben          (z.B. 1 <Kom
ma> 2 <ENTER>)          vorige Eingabe wi
ederholen ->0 eingeben"
```

```
370 FOR n=1 TO f-1:PRINT:PRINT:PRINT "Sp
ieltag";n:PRINT
380 FOR m=1 TO f/2
390 PRINT:PRINT "Spiel";m:INPUT a,b:IF a
>f OR b>f OR a=b THEN GOTO 390 ELSE IF a
=0 OR b=0 THEN m=m-1:GOTO 390 ELSE k(n,m
,1)=a:k(n,m,2)=b:a=0:b=0:PRINT a$(k(n,m,
1));" - ";a$(k(n,m,2))
400 NEXT m,n
410 GOTO 530
420 ON f GOSUB 440,440,440,440,440,450,4
40,460,440,470,440,480,440,490,440,500,4
40,510,440,520
430 GOTO 530
440 RESTORE 5000:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
450 RESTORE 5040:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
460 RESTORE 5100:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
470 RESTORE 5180:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
480 RESTORE 5280:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
490 RESTORE 5400:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
500 RESTORE 5540:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
510 RESTORE 5700:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
```

```
520 RESTORE 5880:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
530 MODE 1:PEN effekt:LOCATE 7,7:PRINT "
Sportverein-Tabellen";:PEN normal
540 LOCATE 7,10
550 PRINT "1.Spieltag eingeben"
560 LOCATE 7,11
570 PRINT "2.aktuelle Tabelle zeigen"
580 LOCATE 7,12
590 PRINT "3.Spieltag ansehen"
600 LOCATE 7,14
610 INPUT "Ihre Wahl (1 bis 3) ";f$
620 IF VAL(f$)<1 OR VAL(f$)>4 THEN GOSUB
 1270:GOSUB 1290:GOTO 420
630 ON VAL(f$) GOSUB 660,900,1010
640 GOTO 530
650 REM Spieltag eingeben
660 MODE 1
670 INPUT "Welcher Spieltag ";f$:IF VAL(
f$)<1 OR VAL(f$)>f*2-2 THEN GOSUB 1290:G
OTO 670
680 IF VAL(f$)>f-1 THEN GOTO 740
690 FOR n=1 TO f/2:PRINT "Spiel";n
700 IF b(VAL(f$),n)<>0 THEN PRINT CHR$(7
);"Fehler! Spielergebnis bekannt:":PRINT
 a$(k(VAL(f$),n,1));"-";a$(k(VAL(f$),n,2
));":";c(k(VAL(f$),n,1),VAL(f$),1);"-";c
(k(VAL(f$),n,1),VAL(f$),2):GOTO 730
710 PRINT a$(k(VAL(f$),n,1));" - ";a$(k(
VAL(f$),n,2));:INPUT a,b:IF a=99 THEN PR
INT "ausgefallen":b(VAL(f$),n)=0:GOTO 73
0 ELSE GOSUB 790
720 b(VAL(f$),n)=1:c(k(VAL(f$),n,1),VAL(
f$),1)=a:c(k(VAL(f$),n,1),VAL(f$),2)=b:c
(k(VAL(f$),n,2),VAL(f$),1)=b:c(k(VAL(f$)
,n,2),VAL(f$),2)=a
730 NEXT n:RETURN
740 FOR n=1 TO f/2:PRINT "Spiel";n
```

```
750 IF b(VAL(f$),n)<>0 THEN PRINT CHR$(7
);"Fehler! Spielergebnis bekannt:":PRINT
 a$(k(VAL(f$)-f+1,n,2));"-";a$(k(VAL(f$)
-f+1,n,1));":";c(k(VAL(f$)-f+1,n,1),VAL(
f$),1);"-";c(k(VAL(f$)-f+1,n,1),VAL(f$),
2):GOTO 780
760 PRINT a$(k(VAL(f$)-f+1,n,2));" - ";a
$(k(VAL(f$)-f+1,n,1));:INPUT a,b:IF a=99
 THEN PRINT "ausgefallen":b(VAL(f$),n)=0
:GOTO 780 ELSE GOSUB 840
770 b(VAL(f$),n)=1:c(k(VAL(f$)-f+1,n,1),
VAL(f$),1)=a:c(k(VAL(f$)-f+1,n,1),VAL(f$
),2)=b:c(k(VAL(f$)-f+1,n,2),VAL(f$),1)=b
:c(k(VAL(f$)-f+1,n,2),VAL(f$),2)=a
780 NEXT n:RETURN
790 e(k(VAL(f$),n,1))=e(k(VAL(f$),n,1))+
a:f(k(VAL(f$),n,1))=f(k(VAL(f$),n,1))+b:
e(k(VAL(f$),n,2))=e(k(VAL(f$),n,2))+b:f(
k(VAL(f$),n,2))=f(k(VAL(f$),n,2))+a
800 IF a=b THEN h(k(VAL(f$),n,1))=h(k(VA
L(f$),n,1))+1:h(k(VAL(f$),n,2))=h(k(VAL(
f$),n,2))+1
810 IF a>b THEN g(k(VAL(f$),n,1))=g(k(VA
L(f$),n,1))+2:i(k(VAL(f$),n,2))=i(k(VAL(
f$),n,2))+2
820 IF a<b THEN i(k(VAL(f$),n,1))=i(k(VA
L(f$),n,1))+2:g(k(VAL(f$),n,2))=g(k(VAL(
f$),n,2))+2
830 RETURN
840 e(k(VAL(f$)-f+1,n,1))=e(k(VAL(f$)-f+
1,n,1))+a:f(k(VAL(f$)-f+1,n,1))=f(k(VAL(
f$)-f+1,n,1))+b:e(k(VAL(f$)-f+1,n,2))=e(
k(VAL(f$)-f+1,n,2))+b:f(k(VAL(f$)-f+1,n,
2))=f(k(VAL(f$)-f+1,n,2))+a
850 IF a=b THEN h(k(VAL(f$)-f+1,n,1))=h(
k(VAL(f$)-f+1,n,1))+1:h(k(VAL(f$)-f+1,n,
2))=h(k(VAL(f$)-f+1,n,2))+1
860 IF a>b THEN g(k(VAL(f$)-f+1,n,1))=g(
k(VAL(f$)-f+1,n,1))+2:i(k(VAL(f$)-f+1,n,
2))=i(k(VAL(f$)-f+1,n,2))+2
```

```
870 IF a<b THEN i(k(VAL(f$)-f+1,n,1))=i(
k(VAL(f$)-f+1,n,1))+2:g(k(VAL(f$)-f+1,n,
2))=g(k(VAL(f$)-f+1,n,2))+2
880 RETURN
890 REM Aktuelle Tabelle zeigen
900 MODE 2:GOSUB 1180
910 PRINT"Pl. Verein          Sp.    +     =
    -      Torverh.    <>    Pkte.       <>
"
920 PRINT STRING$(71,"="):FOR n=1 TO f
930 PRINT n;TAB(5);a$(z5(n));TAB(18);(g(
z5(n))+(h(z5(n))*2)+i(z5(n)))/2;TAB(24);
g(z5(n));TAB(29);h(z5(n));TAB(34);i(z5(n
));TAB(40);e(z5(n));TAB(45);":";f(z5(n))
;
940 PRINT TAB(51);e(z5(n))-f(z5(n));TAB(
57);g(z5(n))+h(z5(n));TAB(61);":";TAB(62
);h(z5(n))+i(z5(n));TAB(68);g(z5(n))-i(z
5(n))
950 NEXT n
960 PRINT STRING$(71,"=")
970 PRINT TAB(21);"<Bitte eine Taste dru
ecken>"
980 a$=INKEY$:IF a$="" THEN GOTO 980
990 RETURN
1000 REM Spieltag ansehen
1010 CLS:PRINT:INPUT "Die ganze Saison (
J/ ) ";f$:f$=UPPER$(f$):IF LEFT$(f$,1)="
J" THEN GOTO 1100
1020 PRINT "Spieltag ( 1 bis";f*2-2;:INP
UT ") ";f$:IF VAL(f$)<1 OR VAL(f$)>f*2-2
 THEN GOSUB 1290:GOTO 1020 ELSE IF VAL(f
$)>f-1 THEN GOTO 1060
1030 n=VAL(f$):CLS:PRINT "Spieltag";n:PR
INT:PRINT:PRINT:FOR m=1 TO f/2:PRINT m;T
AB(5);a$(k(n,m,1));TAB(15);a$(k(n,m,2));
1040 IF b(n,m)=0 THEN PRINT TAB(33);"-
: -": ELSE PRINT TAB(32);c(k(n,m,1),n,1)
;TAB(36);":";TAB(37);c(k(n,m,1),n,2);
```

```
1050 NEXT m:PRINT:PRINT:GOSUB 1270:GOTO
1170
1060 n=VAL(f$):CLS:PRINT "Spieltag";n:n=
n-(f-1):PRINT:PRINT:PRINT:FOR m=1 TO f/2
:PRINT m;TAB(5);a$(k(n,m,2));TAB(14);a$(
k(n,m,1));TAB(23);
1070 IF b(n,m)=1 THEN PRINT c(k(n,m,1),n
,2);TAB(27);":";TAB(28);c(k(n,m,1),n,1);
 ELSE PRINT " -  : -";
1080 IF b(n+(f-1),m)=0 THEN PRINT TAB(33
);"-  : -" ELSE PRINT TAB(32);c(k(n,m,1)
,n+f-1,1);TAB(36);":";TAB(37);c(k(n,m,1)
,n+f-1,2);
1090 NEXT m:PRINT:PRINT:GOSUB 1270:GOTO
1170
1100 FOR n=1 TO f-1:CLS:PRINT "Spieltag"
;n:PRINT:PRINT:PRINT:FOR m=1 TO f/2:PRIN
T m;TAB(5);a$(k(n,m,1));TAB(15);a$(k(n,m
,2));
1110 IF b(n,m)=0 THEN PRINT TAB(33);"-
: -"; ELSE PRINT TAB(32);c(k(n,m,1),n,1)
;TAB(36);":";TAB(37);c(k(n,m,1),n,2);
1120 NEXT m:PRINT:PRINT:GOSUB 1270:NEXT
n
1130 FOR n=1 TO f-1:CLS:PRINT "Spieltag"
;n+f-1:PRINT:PRINT:PRINT:FOR m=1 TO f/2:
PRINT m;TAB(5);a$(k(n,m,2));TAB(14);a$(k
(n,m,1));TAB(23);
1140 IF b(n,m)=0 THEN PRINT " -  : -"; E
LSE PRINT c(k(n,m,1),n,2);TAB(27);":";TA
B(28);c(k(n,m,1),n,1);
1150 IF b(n+f-1,m)=0 THEN PRINT TAB(33);
"-  : -"; ELSE PRINT TAB(32);c(k(n,m,1),
n+f-1,1);TAB(36);":";TAB(37);c(k(n,m,1),
n+f-1,2);
1160 NEXT m:PRINT:PRINT:GOSUB 1270:NEXT
1170 RETURN
1180 REM Sortierunterroutine
1190  FOR n=1 TO f:z1(n)=g(n)+h(n):z2(n)
=e(n)-f(n):z3(n)=e(n):z4(n)=-1:NEXT n
```

```
1200  FOR n=1 TO f
1210  FOR m=1 TO f
1220  IF z1(m)>z4(n) THEN z4(n)=z1(m):a=
m:GOTO 1250
1230  IF z1(m)=z4(n) AND z2(m)>z2(a) THE
N z4(n)=z1(m):a=m:GOTO 1250
1240  IF z1(m)=z4(n) AND z2(m)=z2(a) AND
 z3(m)>z3(a) THEN z4(n)=z1(m):a=m
1250  NEXT m:z1(a)=-1:z5(n)=a:a=f+1:NEXT
 n
1260  RETURN
1270 PRINT:PRINT TAB(7) "<Bitte eine Tas
te druecken>"
1280 a$=INKEY$:IF a$="" THEN GOTO 1280 E
LSE RETURN
1290 PEN effekt:PRINT TAB(12) "Falsche E
ingabe!":PEN normal:GOSUB 1270:RETURN
5000 REM Datas fuer 4 Vereine
5010 DATA 1,4,2,3
5020 DATA 4,3,1,2
5030 DATA 2,4,3,1
5040 REM Datas fuer 6 Vereine
5050 DATA 1,6,2,5,3,4
5060 DATA 6,4,5,3,1,2
5070 DATA 2,6,3,1,4,5
5080 DATA 6,5,1,4,2,3
5090 DATA 3,6,4,2,5,1
5100 REM Datas fuer 8 Vereine
5110 DATA 1,8,2,7,3,6,4,5
5120 DATA 8,5,6,4,7,3,1,2
5130 DATA 2,8,3,1,4,7,5,6
5140 DATA 8,6,7,5,1,4,2,3
5150 DATA 3,8,4,2,5,1,6,7
5160 DATA 8,7,1,6,2,5,3,4
5170 DATA 4,8,5,3,6,2,7,1
5180 REM Datas fuer 10 Vereine
5190 DATA 1,10,2,9,3,8,4,7,5,6
5200 DATA 10,6,7,5,8,4,9,3,1,2
5210 DATA 2,10,3,1,4,9,5,8,6,7
5220 DATA 10,7,8,6,9,5,1,4,2,3
```

```
5230 DATA 3,10,4,2,5,1,6,9,7,8
5240 DATA 10,8,9,7,1,6,2,5,3,4
5250 DATA 4,10,5,3,6,2,7,1,8,9
5260 DATA 10,9,1,8,2,7,3,6,4,5
5270 DATA 5,10,6,4,7,3,8,2,9,1
5280 REM Datas fuer 12 Vereine
5290 DATA 1,12,2,11,3,10,4,9,5,8,6,7
5300 DATA 12,7,8,6,9,5,10,4,11,3,1,2
5310 DATA 2,12,3,1,4,11,5,10,6,9,7,8
5320 DATA 12,8,9,7,10,6,11,5,1,4,2,3
5330 DATA 3,12,4,2,5,1,6,11,7,10,8,9
5340 DATA 12,9,10,8,11,7,1,6,2,5,3,4
5350 DATA 4,12,5,3,6,2,7,1,8,11,9,10
5360 DATA 12,10,11,9,1,8,2,7,3,6,4,5
5370 DATA 5,12,6,4,7,3,8,2,9,1,10,11
5380 DATA 12,11,1,10,2,9,3,8,4,7,5,6
5390 DATA 6,12,7,5,8,4,9,3,10,2,11,1
5400 REM Datas fuer 14 Vereine
5410 DATA 1,14,2,13,3,12,4,11,5,10,6,9,7
,8
5420 DATA 14,8,9,7,10,6,11,5,12,4,13,3,1
,2
5430 DATA 2,14,3,1,4,13,5,12,6,11,7,10,8
,9
5440 DATA 14,9,10,8,11,7,12,6,13,5,1,4,2
,3
5450 DATA 3,14,4,2,5,1,6,13,7,12,8,11,9,
10
5460 DATA 14,10,11,9,12,8,13,7,1,6,2,5,3
,4
5470 DATA 4,14,5,3,6,2,7,1,8,13,9,12,10,
11
5480 DATA 14,11,12,10,13,9,1,8,2,7,3,6,4
,5
5490 DATA 5,14,6,4,7,3,8,2,9,1,10,13,11,
12
5500 DATA 14,12,13,11,1,10,2,9,3,8,4,7,5
,6
5510 DATA 6,14,7,5,8,4,9,3,10,2,11,1,12,
13
```

```
5520 DATA 14,13,1,12,2,11,3,10,4,9,5,8,6
,7
5530 DATA 7,14,8,6,9,5,10,4,11,3,12,2,13
,1
5540 REM Datas fuer 16 Vereine
5550 DATA 1,16,2,15,3,14,4,13,5,12,6,11,
7,10,8,9
5560 DATA 16,9,10,8,11,7,12,6,13,5,14,4,
15,3,1,2
5570 DATA 2,16,3,1,4,15,5,14,6,13,7,12,8
,11,9,10
5580 DATA 16,10,11,9,12,8,13,7,14,6,15,5
,1,4,2,3
5590 DATA 3,16,4,2,5,1,6,15,7,14,8,13,9,
12,10,11
5600 DATA 16,11,12,10,13,9,14,8,15,7,1,6
,2,5,3,4
5610 DATA 4,16,5,3,6,2,7,1,8,15,9,14,10,
13,11,12
5620 DATA 16,12,13,11,14,10,15,9,1,8,2,7
,3,6,4,5
5630 DATA 5,16,6,4,7,3,8,2,9,1,10,15,11,
14,12,13
5640 DATA 16,13,14,12,15,11,1,10,2,9,3,8
,4,7,5,6
5650 DATA 6,16,7,5,8,4,9,3,10,2,11,1,12,
15,13,14
5660 DATA 16,14,15,13,1,12,2,11,3,10,4,9
,5,8,6,7
5670 DATA 7,16,8,6,9,5,10,4,11,3,12,2,13
,1,14,15
5680 DATA 16,15,1,14,2,13,3,12,4,11,5,10
,6,9,7,8
5690 DATA 8,16,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1
5700 REM Datas fuer 18 Vereine
5710 DATA 1,18,2,17,3,16,4,15,5,14,6,13,
7,12,8,11,9,10
5720 DATA 18,10,11,9,12,8,13,7,14,6,15,5
,16,4,17,3,1,2
```

```
5730 DATA 2,18,3,1,4,17,5,16,6,15,7,14,8
,13,9,12,10,11
5740 DATA 18,11,12,10,13,9,14,8,15,7,16,
6,17,5,1,4,2,3
5750 DATA 3,18,4,2,5,1,6,17,7,16,8,15,9,
14,10,13,11,12
5760 DATA 18,12,13,11,14,10,15,9,16,8,17
,7,1,6,2,5,3,4
5770 DATA 4,18,5,3,6,2,7,1,8,17,9,16,10,
15,11,14,12,13
5780 DATA 18,13,14,12,15,11,16,10,17,9,1
,8,2,7,3,6,4,5
5790 DATA 5,18,6,4,7,3,8,2,9,1,10,17,11,
16,12,15,13,14
5800 DATA 18,14,15,13,16,12,17,11,1,10,2
,9,3,8,4,7,5,6
5810 DATA 6,18,7,5,8,4,9,3,10,2,11,1,12,
17,13,16,14,15
5820 DATA 18,15,16,14,17,13,1,12,2,11,3,
10,4,9,5,8,6,7
5830 DATA 7,18,8,6,9,5,10,4,11,3,12,2,13
,1,14,17,15,16
5840 DATA 18,16,17,15,1,14,2,13,3,12,4,1
1,5,10,6,9,7,8
5850 DATA 8,18,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1,16,17
5860 DATA 18,17,1,16,2,15,3,14,4,13,5,12
,6,11,7,10,8,9
5870 DATA 9,18,10,8,11,7,12,6,13,5,14,4,
15,3,16,2,17,1
5880 REM Datas fuer 20 Vereine
5890 DATA 1,20,2,19,3,18,4,17,5,16,6,15,
7,14,8,13,9,12,10,11
5900 DATA 20,11,12,10,13,9,14,8,15,7,16,
6,17,5,18,4,19,3,1,2
5910 DATA 2,20,3,1,4,19,5,18,6,17,7,16,8
,15,9,14,10,13,11,12
5920 DATA 20,12,13,11,14,10,15,9,16,8,17
,7,18,6,19,5,1,4,2,3
```

```
5930 DATA 3,20,4,2,5,1,6,19,7,18,8,17,9,
16,10,15,11,14,12,13
5940 DATA 20,13,14,12,15,11,16,10,17,9,1
8,8,19,7,1,6,2,5,3,4
5950 DATA 4,20,5,3,6,2,7,1,8,19,9,18,10,
17,11,16,12,15,13,14
5960 DATA 20,14,15,13,16,12,17,11,18,10,
19,9,1,8,2,7,3,6,4,5
5970 DATA 5,20,6,4,7,3,8,2,9,1,10,19,11,
18,12,17,13,16,14,15
5980 DATA 20,15,16,14,17,13,18,12,19,11,
1,10,2,9,3,8,4,7,5,6
5990 DATA 6,20,7,5,8,4,9,3,10,2,11,1,12,
19,13,18,14,17,15,16
6000 DATA 20,16,17,15,18,14,19,13,1,12,2
,11,3,10,4,9,5,8,6,7
6010 DATA 7,20,8,6,9,5,10,4,11,3,12,2,13
,1,14,19,15,18,16,17
6020 DATA 20,17,18,16,19,15,1,14,2,13,3,
12,4,11,5,10,6,9,7,8
6030 DATA 8,20,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1,16,19,17,18
6040 DATA 20,18,19,17,1,16,2,15,3,14,4,1
3,5,12,6,11,7,10,8,9
6050 DATA 9,20,10,8,11,7,12,6,13,5,14,4,
15,3,16,2,17,1,18,19
6060 DATA 20,19,1,18,2,17,3,16,4,15,5,14
,6,13,7,12,8,11,9,10
6070 DATA 10,20,11,9,12,8,13,7,14,6,15,5
,16,4,17,3,18,2,19,1
6080 REM Datas fuer 2.Liga 1984/85
         Beispiel fuer 20 Vereine
6090 DATA Hannover,Aachen,Hertha,Wattens
.,Saarbru.
6100 DATA Solingen,Kassel,Nuernbe.,Hombu
rg,Buersta.
6110 DATA Offenba.,Ulm,Koeln,Freiburg,Da
rmsta.
6120 DATA Oberhau.,Stuttga.,BW 90,St.Pau
li,Duisburg
```

```
6130 REM Datas fuer Bundesliga 1984/85
         Beispiel fuer 18 Vereine
6140 DATA Muenchen,Gladbach,Werder,Bochu
m,Hamburg,Kaisers.
6150 DATA Koeln,Stuttga.,Uerding.,Leverk
u.,Karlsru.,Frankfu.
6160 DATA Waldhof,Schalke,Duessel.,Biele
fe.,Dortmund,Braunsc.
6170 REM Datas fuer 16 Vereine
6180 DATA A,B,C,D,E,F,G,H
6190 DATA I,J,K,L,M,N,O,P
6200 REM Datas fuer 14 Vereine
6210 DATA A,B,C,D,E,F,G
6220 DATA H,I,J,K,L,M,N
6230 REM Datas fuer 12 Vereine
6240 DATA A,B,C,D,E,F
6250 DATA G,H,I,J,K,L
6260 REM Datas fuer 10 Vereine
6270 DATA A,B,C,D,E
6280 DATA F,G,H,I,J
6290 REM Datas fuer 8 Vereine
6300 DATA A,B,C,D
6310 DATA E,F,G,H
6320 REM Datas fuer 6 Vereine
6330 DATA A,B,C
6340 DATA D,E,F
6350 REM Datas fuer 4 Vereine
6360 DATA A,B
6370 DATA C,D
```

Pl.	Verein	Sp.	+	=	-	Torverh.	<>	Pkte.	<>
1	Darmsta.	2	4	0	0	10 : 5	5	4 : 0	4
2	Kassel	2	4	0	0	10 : 7	3	4 : 0	4
3	Offenba.	2	4	0	0	7 : 4	3	4 : 0	4
4	Hannover	2	2	1	0	8 : 2	6	3 : 1	2
5	Nuernbe.	2	2	1	0	12 : 10	2	3 : 1	2
6	BW 90	2	2	1	0	8 : 7	1	3 : 1	2
7	Ulm	2	2	1	0	5 : 4	1	3 : 1	2
8	Aachen	2	0	2	0	5 : 5	0	2 : 2	0
9	St.Pauli	2	0	2	0	5 : 5	0	2 : 2	0
10	Wattens.	1	0	1	0	4 : 4	0	1 : 1	0
11	Stuttga.	1	0	1	0	4 : 4	0	1 : 1	0
12	Saarbru.	1	0	1	0	3 : 3	0	1 : 1	0
13	Oberhau.	1	0	1	0	3 : 3	0	1 : 1	0
14	Hertha	2	0	1	2	7 : 8	-1	1 : 3	-2
15	Homburg	2	0	1	2	4 : 5	-1	1 : 3	-2
16	Koeln	2	0	1	2	10 : 12	-2	1 : 3	-2
17	Duisburg	2	0	1	2	2 : 8	-6	1 : 3	-2
18	Freiburg	2	0	0	4	7 : 10	-3	0 : 4	-4
19	Buersta.	2	0	0	4	4 : 7	-3	0 : 4	-4
20	Solingen	2	0	0	4	5 : 10	-5	0 : 4	-4

<Bitte eine Taste druecken>

Knobeln
=======

Es ist anzunehmen, wie leider bei so vielen neuen Homecomputern immer wieder zu beoachten, daß der CPC in der Anfangszeit großenteils mit Spielsoftware ausgerüstet wird. Wir haben in dieser Programmsammlung vielmehr den Versuch unternommen, aufzuzeigen, wie man auch etwas sinnvolles mit einem Homecomputer machen oder aber ihn ein klein wenig genauer kennenleren kann.

Jedoch soll es auch nicht an Spielerei mangeln. Zu diesem Zweck haben wir ein sehr ausführiches Knobel- oder Yahtzee-Programm in die Programmsammlung aufgenommen.

Hier werden die Fähigkeiten des CPC-BASIC in die Tat umgesetzt; so haben wir es gleich mit vier verschiedenen Bildschirmfenstern zu tun, so daß eine leichte Editierung der Eingaben direkt mit den Pfeiltasten auf dem Bildschirm sichtbar vorgenommen werden kann.

Die Regeln dieses Würfelspiels sind hoffentlich bekannt (jeder muß mit 5 Würfeln versuchen, die 12 vorgegebenen Bedingungen zu erfüllen), sonst schauen Sie bitte mal in einem Spielebuch nach oder knobeln sich die Gebrauchsanweisung selbst durch oftmaliges Spielen zusammen. Hier nur soviel: Sie sind 12 mal an der Reihe und müssen mit jeweils 3 Würfen versuchen, nach der vorgegebenen Punktezählung am Ende besonders gut auszusehen, d.h. möglichst viele Punkte zu erzielen.

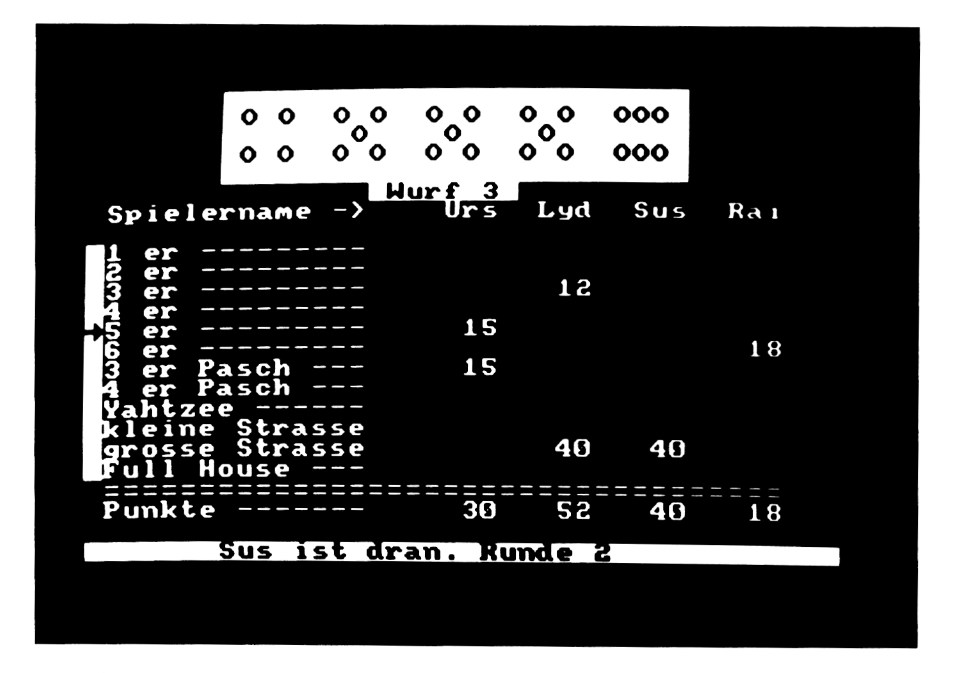
Der CPC würfelt nicht nur für Sie, er ordnet die Würfel auch in aufsteigender Reihenfolge. Mit den Cursor- und der COPY-Taste suchen Sie sich die Wüfel heraus, die ein weiteres mal dem Gesetz des Zufalls gehorchen sollen. Ist dreimal gewürfelt worden, suchen Sie sich mit Hilfe der Cursor- und der (ENTER)-Taste die Zeile in der Tabelle heraus, in die ihr hoffentlich gutes Ergebnis schließlich landen soll (viele 6 er - Eintrag in das Feld '6 er' ...). Jedes Mal wird zudem die erreichte Gesamtpunktzahl angezeigt. Viel Spaß!

Wurf 2

Spielername -> Urs Lyd Sus Rai

1 er ----------
2 er ----------
3 er ----------
4 er ----------
5 er ----------
6 er ----------
3 er Pasch ---
4 er Pasch ---
Yahtzee ------
kleine Strasse
grosse Strasse
Full House ---
==
Punkte --------

Urs ist dran. Runde 1



```
10 REM Knobeln
20 REM CPC Basic Programme
30 REM Copyright 1984 DATA BECKER &
                        Rainer Lueers
40 INK Ø,1:INK 1,24:INK 2,24
50 REM Vorbedingungen fuer
       Programmablauf sicherstellen
60 MODE 1
70 INPUT "Anzahl der Spieler (1 bis 4) "
;spieler
80 IF spieler<1 OR spieler>4 THEN GOSUB
2350:GOTO 60
90 DIM spiel(spieler,12),spiel1(spieler,
12)
100 FOR n=1 TO spieler
110 PRINT:PRINT "Spieler";n;":";
120 INPUT spieler$(n)
130 IF LEN(spieler$(n))>3 THEN spieler$(
n)=LEFT$(spieler$(n),3)
140 IF LEN(spieler$(n))=2 THEN spieler$(
n)=spieler$(n)+" "
150 IF LEN(spieler$(n))=1 THEN spieler$(
n)=spieler$(n)+"  "
160 IF LEN(spieler$(n))=0 THEN GOSUB 235
0:RUN
170 NEXT n
180 MODE 1
190 REM Fenster fuer Wuerfeldarstellung
200 WINDOW #1,8,32,1,6
210 REM Fenster fuer Ergebnistabelle
220 WINDOW #2,2,40,7,22
230 REM Fenster fuer Informationsbalken
240 WINDOW #3,1,40,24,24
250 REM Fenster fuer Auswahl zur
        Tabelle
260 WINDOW #4,1,1,9,21
270 REM Festlegen von Untergrund- und
        Schriftfarbe in den
        verschiedenen Fenstern
```

```
280 PAPER #4,2:PEN #4,0:CLS #4:PAPER #3,
2:PEN #3,0:CLS #4:CLS #3:PAPER #1,2:PEN
#1,0:CLS #1:PAPER #2,0:PEN #2,1:CLS #2:L
OCATE #4,1,13:PRINT #4,CHR$(143);
290 REM Bildschirmaufbau der Tabelle
300 PRINT#2,"Spielername ->     ";:FOR n=
1 TO spieler:PRINT#2,spieler$(n);"  ";:N
EXT n:PRINT#2:PRINT#2:PRINT#2,"1 er ----
-----":PRINT #2,"2 er ---------":PRINT#2
,"3 er ---------":PRINT#2,"4 er --------
-":PRINT #2,"5 er ---------":PRINT#2,"6
er ---------"
310 PRINT#2,"3 er Pasch ---":PRINT#2,"4
er Pasch ---":PRINT#2,"Yahtzee ------":P
RINT #2,"kleine Strasse":PRINT #2,"gross
e Strasse":PRINT #2,"Full House ---":PRI
NT #2,STRING$(36,"="):PRINT #2,"Punkte -
------"
320 REM Einlesen der Wuerfeldarstellung
        aus DATA-Zeilen
330 RESTORE:FOR n=1 TO 6:FOR m=1 TO 3
340 READ a$(n,m):NEXT m,n
350 REM Beginn des Hauptprogramms:
        Ein Spiel mit zwoelf Runden
360 FOR runde=1 TO 12
370 FOR runde1=1 TO spieler
380 CLS #3
390 LOCATE #3,8,1:PRINT #3,spieler$(rund
e1);" ist dran. Runde";runde;
400 REM Fuenf Zufallszahlen zwischen
        eins und sechs erzeugen
410 FOR n=1 TO 5
420 RANDOMIZE TIME
430 a=INT(10*(RND(TIME)))
440 IF a<1 OR a>6 THEN GOTO 420
450 zahl(n)=a
460 REM Darstellung des entsprechenden
        Wuerfelaussehens
470 FOR m=2 TO 4
480 LOCATE #1,n*5-3,m
```

```
490 PRINT #1,a$(a,m-1);
500 NEXT m:NEXT n
510 REM Einsprung in
        Wuerfelsortierroutine
520 GOSUB 950
530 REM Jeder Spieler hat drei
        Durchgaenge in einer Runde
540 FOR zaehler=1 TO 2
550 REM Einsprung in Aenderungsroutine
560 GOSUB 820
570 CLS #1
580 FOR n=1 TO 5
590 REM Fuenf Zufallszahlen zwischen
        eins und sechs erzeugen
600 IF wa(n)=1 THEN RANDOMIZE TIME:a=INT
(10*(RND(TIME))):IF a<1 OR a>6 THEN GOTO
 600 ELSE zahl(n)=a ELSE a=zahl(n):GOTO
620
610 REM Darstellung des
        entsprechenden
        Wuerfelaussehens
620 wa(n)=0
630 FOR m=2 TO 4
640 LOCATE #1,n*5-3,m
650 PRINT #1,a$(a,m-1);
660 NEXT m
670 NEXT n
680 REM Einsprung in
        Wuerfelsortierroutine
690 GOSUB 950
700 NEXT zaehler:zaehler=0
710 REM Was soll mit dem erzielten
        Wurf gemacht werden?
720 GOSUB 1420
730 NEXT runde1,runde
740 REM Spielende und Frage:
        Nochmal?
750 a$=INKEY$
760 IF a$="" THEN GOTO 750
770 CLS
```

```
780 INPUT "Nochmal ( /N) ";a$
790 a$=UPPER$(a$)
800 IF LEFT$(a$,1)="N" THEN END ELSE RUN
810 REM Aenderungsroutine
820 LOCATE #1,3,5:PRINT #1,CHR$(244);
830 a$=INKEY$
840 IF a$="" THEN GOTO 830
850 REM Pfeiltaste nach 'links'
         gedrueckt
860 IF a$=CHR$(242) AND POS(#1)>4 THEN L
OCATE #1,POS(#1)-1,5:PRINT #1," ";:LOCAT
E #1,POS(#1)-6,5:PRINT #1,CHR$(244);:GOT
O 830
870 REM Pfeiltaste nach 'rechts'
         gedrueckt
880 IF a$=CHR$(243) AND POS(#1)<24 THEN
LOCATE #1,POS(#1)-1,5:PRINT #1," ";:LOCA
TE #1,POS(#1)+4,5:PRINT #1,CHR$(244);:GO
TO 830
890 REM COPY-Taste gedrueckt
         (Wert = umgekehrt der Anzeige)
900 IF a$<>CHR$(224) THEN GOTO 920 ELSE
IF wa((POS(#1)+1)/5)=1 THEN wa((POS(#1)/
+1)/5)=0:PEN #1,2:LOCATE #1,POS(#1)-1,1:
PRINT #1," ";:PEN #1,0:LOCATE #1,POS(#1)
,5 ELSE wa((POS(#1)+1)/5)=1:LOCATE #1,PO
S(#1)-1,1:PRINT #1,CHR$(245);:LOCATE #1,
POS(#1),5
910 REM <ENTER>-Taste gedrueckt
920 IF a$=CHR$(13) THEN RETURN
930 GOTO 830
940 REM Wurfsortierroutinen
950 FOR nn=1 TO 5
960 z2(nn)=7
970 NEXT nn
980 FOR nn=1 TO 5
990 FOR mm=1 TO 5
1000 IF zahl(mm)<=z2(nn) THEN z2(nn)=zah
l(mm):z1(nn)=wa(mm):merker=mm
1010 NEXT mm
```

```
1020 zahl(merker)=7
1030 NEXT nn
1040 FOR nn=1 TO 5
1050 zahl(nn)=z2(nn)
1060 wa(nn)=z1(nn)
1070 NEXT nn
1080 CLS #1
1090 FOR nn=1 TO 5
1100 FOR mm=2 TO 4
1110 LOCATE #1,nn*5-3,mm:PRINT #1,a$(zah
l(nn),mm-1);
1120 NEXT mm
1130 NEXT nn
1140 LOCATE #1,1,6:PRINT #1,STRING$(8,CH
R$(143));" Wurf";zaehler+1;STRING$(9,CHR
$(143));
1150 RETURN
1160 REM Aussehen der Wuerfel mit
         der entsprechenden Augenzahl
         (eins bis sechs) in
         DATA-Zeilen
1170 REM Wuerfelzahl 1
1180 DATA "   "
1190 DATA " O "
1200 DATA "   "
1210 REM Wuerfelzahl 2
1220 DATA "  O"
1230 DATA "   "
1240 DATA "O  "
1250 REM Wuerfelzahl 3
1260 DATA "  O"
1270 DATA " O "
1280 DATA "O  "
1290 REM Wuerfelzahl 4
1300 DATA "O O"
1310 DATA "   "
1320 DATA "O O"
1330 REM Wuerfelzahl 5
1340 DATA "O O"
1350 DATA " O "
```

```
1360 DATA "O O"
1370 REM Wuerfelzahl 6
1380 DATA "OOO"
1390 DATA "   "
1400 DATA "OOO"
1410 REM Eingaberoutine fuer
         Tabellenwerte
1420 CLS #4
1430 LOCATE #4,1,13
1440 PRINT #4,CHR$(143);
1450 LOCATE #4,1,1
1460 PRINT #4,CHR$(243);
1470 a$=INKEY$
1480 IF a$="" THEN GOTO 1470
1490 REM Pfeiltaste nach 'unten'
         gedrueckt
1500 IF a$=CHR$(241) AND VPOS(#4)<13 THE
N LOCATE #4,1,VPOS(#4)-1 ELSE GOTO 1550
1510 PRINT #4," ";
1520 LOCATE #4,1,VPOS(#4)
1530 PRINT #4,CHR$(243);
1540 REM Pfeiltaste nach 'oben'
         gedrueckt
1550 IF a$=CHR$(240) AND VPOS(#4)>2 THEN
 LOCATE #4,1,VPOS(#4)-1 ELSE GOTO 1600
1560 PRINT #4," ";
1570 LOCATE #4,1,VPOS(#4)-2
1580 PRINT #4,CHR$(243);
1590 REM <ENTER>-Taste gedrueckt.
         Wenn Wert noch nicht
         eingtragen, so wird er vermerkt
         und die Punktzahl gemaess
         Unterprogramm errechnet
1600 IF a$=CHR$(13) THEN IF spiel1(runde
1,VPOS(#4)-1)<>1 THEN spiel1(runde1,VPOS
(#4)-1)=1:GOSUB 1820 ELSE GOTO 1470 ELSE
 GOTO 1470
1610 spiel(runde1,runde)=ergebnis
1620 CLS #3
1630 LOCATE #3,9,1
```

```
1640 PRINT #3,"Spieler ";spieler$(runde1
);":";ergebnis;"Punkte";
1650 REM Eintragung des Ergebnisses in
         die Tabelle (an der
         entsprechenden Stelle)
1660 LOCATE #2,14+runde1*5,VPOS(#4)+1
1670 PRINT #2,ergebnis;
1680 LOCATE #2,14+runde1*5,16
1690 FOR n=1 TO 12
1700 endergebnis(runde1)=endergebnis(run
de1)+spiel(runde1,n)
1710 NEXT n
1720 PRINT #2,endergebnis(runde1);
1730 endergebnis(runde1)=0
1740 a$=INKEY$
1750 IF a$="" THEN GOTO 1740
1760 CLS #1
1770 CLS #4
1780 LOCATE #4,1,13
1790 PRINT #4,CHR$(143);
1800 CLS #3
1810 RETURN
1820 ergebnis=0
1830 REM Unterprogramme zur Errechnung
         der erzielten Punktzahl
1840 ON VPOS(#4)-1 GOSUB 1870,1920,1970,
2020,2070,2120,2170,2200,2230,2260,2300,
2330
1850 RETURN
1860 REM 1 er Wurf gezaehlt
         (Addition aller 1 er)
1870 FOR n=1 TO 5
1880 IF zahl(n)=1 THEN ergebnis=ergebnis
+1
1890 NEXT n
1900 RETURN
1910 REM 2 er Wurf gezaehlt
         (Addition aller 2 er)
1920 FOR n=1 TO 5
```

```
1930 IF zahl(n)=2 THEN ergebnis=ergebnis
+2
1940 NEXT n
1950 RETURN
1960 REM 3 er Wurf gezaehlt
        (Addition aller 3 er)
1970 FOR n=1 TO 5
1980 IF zahl(n)=3 THEN ergebnis=ergebnis
+3
1990 NEXT n
2000 RETURN
2010 REM 4 er Wurf gezaehlt
        (Addition aller 4 er)
2020 FOR n=1 TO 5
2030 IF zahl(n)=4 THEN ergebnis=ergebnis
+4
2040 NEXT n
2050 RETURN
2060 REM 5 er Wurf gezaehlt
        (Addition aller 5 er)
2070 FOR n=1 TO 5
2080 IF zahl(n)=5 THEN ergebnis=ergebnis
+5
2090 NEXT n
2100 RETURN
2110 REM 6 er Wurf gezaehlt
        (Addition aller 6 er)
2120 FOR n=1 TO 5
2130 IF zahl(n)=6 THEN ergebnis=ergebnis
+6
2140 NEXT n
2150 RETURN
2160 REM 3 er Pasch-Pruefung;
        3 er Pasch = alle Wuerfelaugen
2170 IF (zahl(1)=zahl(2) AND zahl(2)=zah
l(3)) OR (zahl(2)=zahl(3) AND zahl(3)=za
hl(4)) OR (zahl(3)=zahl(4) AND zahl(4)=z
ahl(5)) THEN FOR n=1 TO 5:ergebnis=ergeb
nis+zahl(n):NEXT n
2180 RETURN
2190 REM 4 er Pasch-Pruefung;
```

```
2200 IF (zahl(1)=zahl(2) AND zahl(2)=zah
l(3) AND zahl(3)=zahl(4)) OR (zahl(2)=za
hl(3) AND zahl(3)=zahl(4) AND zahl(4)=za
hl(5)) THEN FOR n=1 TO 5:ergebnis=ergebn
is+zahl(n):NEXT n
2210 RETURN
2220 REM 5 er Pasch = Yahtzee-Pruefung
         Yahtzee = 50 Punkte
2230 IF zahl(1)=zahl(2) AND zahl(2)=zahl
(3) AND zahl(3)=zahl(4) AND zahl(4)=zahl
(5) THEN ergebnis=50
2240 RETURN
2250 REM kleine Strasse-Pruefung;
         kleine Strasse = 30 Punkte
2260 IF (zahl(1)=zahl(2)-1 AND zahl(2)=z
ahl(3)-1) OR (zahl(2)=zahl(3)-1 AND zahl
(3)=zahl(4)-1) OR (zahl(3)=zahl(4)-1 AND
 zahl(4)=zahl(5)-1) OR (zahl(1)=zahl(3)-
1 AND zahl(2)=zahl(4)-1) OR (zahl(1)=zah
l(4)-1 AND zahl(2)=zahl(5)-1) THEN ergeb
nis=30
2270 IF zahl(2)=zahl(4)-1 AND zahl(4)=za
hl(5)-1 THEN ergebnis=30
2280 RETURN
2290 REM grosse Strasse-Pruefung;
         grosse Strasse = 40 Punkte
2300 IF (zahl(1)=zahl(2)-1 AND zahl(2)=z
ahl(3)-1 AND zahl(3)=zahl(4)-1) OR (zahl
(2)=zahl(3)-1 AND zahl(3)=zahl(4)-1 AND
zahl(4)=zahl(5)-1) OR (zahl(1)=zahl(3)-1
 AND zahl(3)=zahl(4)-1 AND zahl(4)=zahl(
5)-1) THEN ergebnis=40
2310 RETURN
2320 REM Full House-Pruefung;
         Full House = 25 Punkte
2330 IF (zahl(1)=zahl(2) AND zahl(3)=zah
l(4) AND zahl(4)=zahl(5)) OR (zahl(1)=za
hl(2) AND zahl(2)=zahl(3) AND zahl(4)=za
hl(5)) THEN ergebnis=25
2340 RETURN
2350 a$=INKEY$:IF a$="" THEN GOTO 2350 E
LSE PRINT ASC(a$):GOTO 2350
```

Codeknacker

Dieses Spiel gibt es in mannigfaltigen Versionen z.B. unter den Namen 'Superhirn' und 'Mastermind'.

Wir haben das Computerprogramm jedoch umbenannt in 'Codeknakker', weil es sich von den Vorbildern doch in gewisser Weise stark unterscheidet.

Zu Beginn können Sie wählen, aus wieviel Farben=Zahlen die zu erratende Folge zusammengesetzt sein soll. Es stehen bis zu acht Farben=Zahlen zur Verfügung.

Anschließend erfolgt eine zweite Wahl: Aus wieviel Teilen soll die Folge bestehen? Es ist zwar möglich mehr Farben als Teile zu bestimmen, allerdings ist das umgekehrte Verhältnis nicht vorgesehen und wird somit ignoriert. Dies hat folgenden Grund: Der Spieler soll zu Beginn wissen, mit welcher Planung er es zu tun hat; so ist es klar, daß nach der Auswahl von jeweils acht Farben und acht Stellen auch wahrlich acht Farben in der Folge auftreten; Doppel oder gar Dreier sind somit ausgeschlossen!

Aus diesem Grund wird während des Spielablaufs ähnlich weiterverfahren: es nützt nicht, durch Vorgabe z.B. der Zahlenkombination '11112222' herausfinden zu wollen, ob die '1' bzw. '2' gleich mehrfach und an den entsprechenden Stellen in der Folge vorkommt.

Es wird sowohl auf dem Bildschirm angezeigt, wie viele Farben richtig aber auch wie viele gar genau an der richtigen Stelle positioniert wurden. Für den 'Codeknacker'-Anfänger ein recht schwieriges Unterfangen. Aber Übung macht den Meister. Um den Überblick nicht zu verlieren, wird der Bildschirm nach jeder Eingabe hochgescrollt (die letzten vier bis fünf Eintragungen bleiben oben sichtbar). Zudem werden alle Ergebnisse und Zahleingaben festgehalten. Man kann die gesamte Eingabefolge durch Drücken der (ENTER)-Taste wieder auf den Bildschirm bringen. Nun aber genug der Vorstellung. Viel Spaß!

```
Eingabe (Dez.,Hex o. Bin.) ? d255
dezimal   binaer oder dual    hexadezimal
------------------------------------------
255       00000000 11111111      00FF

Eingabe (Dez.,Hex o. Bin.) ? hffff
dezimal   binaer oder dual    hexadezimal
------------------------------------------
 65535    11111111 11111111      FFFF

Eingabe (Dez.,Hex o. Bin.) ? b111
dezimal   binaer oder dual    hexadezimal
------------------------------------------
 7        00000000 00000111      0007

Eingabe (Dez.,Hex o. Bin.) ? █
```

```
Umrechnung DEZ. - 5 ersystem
==============================

Ihre Eingabe (DEZ.) ? 230

Anzahl: 46     MOD: 0
Anzahl: 9      MOD: 1
Anzahl: 1      MOD: 4

 1410


Welches Zahlsystem (2 bis 9) ? █
```

```
10 REM Codeknacker
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                        Rainer Lueers
40 REM Dimensionierung und Abfrage
       der Vorbedingungen
50 DIM merk$(100),schwarz(100),weiss(100
)
60 PAPER 0:PEN 1:MODE 1
70 INPUT "Anzahl der Farben (1 bis 8) ";
f$
80 IF VAL(f$)<1 OR VAL(f$)>8 THEN GOSUB
460:GOTO 70
90 INPUT "Anzahl der Stellen (1 bis 8) "
;f1$
100 IF VAL(f1$)<1 OR VAL(f$)>8 THEN GOSU
B 460:GOTO 90
110 IF VAL(f1$)>VAL(f$) THEN GOSUB 460:G
OTO 90
120 REM Auswahl der Zahlen durch die
        RND=Zufallsfunktion
130 FOR n=1 TO VAL(f1$)
140 a=INT(10*RND(TIME))
150 IF a<1 OR a>VAL(f$) THEN GOTO 140
160 REM Alle ausgewaehlten Zahlen
        sollen unterschiedlich sein
170 FOR m=1 TO n-1
180 a(m)=VAL(MID$(a$,m,1))
190 NEXT m
200 FOR m=1 TO n-1
210 IF a=a(m) THEN GOTO 140 ELSE NEXT m
220 a$=a$+RIGHT$(STR$(a),1)
230 NEXT n
240 FOR n=1 TO VAL(f1$)
250 a(n)=VAL(MID$(a$,n,1))
260 NEXT n
270 CLS
280 REM Eingaberoutine
```

```
290 PRINT "Eine Zahl mit";VAL(f1$);"Ziff
ern ";:INPUT f3$
300 REM Auflisung aller Zahlen, die
        bisher eingegeben wurden
310 IF f3$="" THEN CLS:FOR z1=1 TO z:PRI
NT "Zug"z1;TAB(8);merk$(z1);"    OK:";sc
hwarz(z1);"  not OK:";weiss(z1):NEXT z1:
PRINT:GOTO 290  ELSE IF LEN(f3$)<>VAL(f1
$) THEN GOSUB 460:GOTO 290
320 FOR n=1 TO VAL(f1$)
330 b(n)=VAL(MID$(f3$,n,1))
340 IF a(n)=b(n) THEN schwarz=schwarz+1:
c(n)=-1
350 NEXT n
360 FOR m=1 TO VAL(f1$)
370 FOR n=1 TO VAL(f1$)
380 IF a(n)=b(m) AND n<>m THEN weiss=wei
ss+1:GOTO 410
390 NEXT n
400 REM Ausgaberoutine und
        anschliessende Ueberpruefung
        auf Richtigkeit
410 NEXT m:PRINT "Farben an der falschen
 Stelle :";weiss
420 z=z+1:merk$(z)=f3$:weiss(z)=weiss:sc
hwarz(z)=schwarz:weiss=0
430 PRINT "Farben an der richtigen Stell
e:";schwarz
440 IF schwarz=VAL(f1$) THEN PRINT:PAPER
 1:PEN 0:PRINT " Genau richtig in";z;"Zu
egen";:PAPER 0:PEN 1:PRINT:PRINT:INPUT "
Nochmal (J/ ) ";f$:IF UPPER$(LEFT$(f$,1)
)="J" THEN RUN ELSE END ELSE schwarz=0:G
OTO 290
450 GOTO 290
460 PRINT:PRINT "Falsche Eingabe!":PRINT
:RETURN
```

Reaktion

Um Leute zu überraschen, vielleicht auch, um Leute, die bisher der ganzen Computerei eher negativ gegenüberstehen, sei dieses Programm zur Vorführung dringendst empfohlen.

Der ganze Inhalt, was die Programmierung an und für sich betrifft: Es muß eine Taste so schnell wie nur irgendwie möglich gedrückt werden, wenn der Computer dazu auffordert. Im Nachhinein erfolgt eine Wertung und es wird ein weiteres Mal zu einem neuen Spiel mit bis zu vier Teilnehmern aufgefordert.

Der Clou dieses Programms liegt vielmehr darin, wie mit dem nichts ahnenden Computerbediener umgegangen wird. Wie bereits an anderer Stelle in diesem Buch gesagt: Die Computer sind alle sehr dumm. Sie werden erst durch die Programmierung von uns Menschen schlau gemacht. Gerade dies wird dem laienhaften Benutzer beim Ablauf dieses Programms klar gemacht:

Erfolgt zu Beginn keine Namenseingabe, wird der CPC bzw. dieses Programm langsam aber sicher richtig knartschig ... bis er gar droht, man mache den Computer durch Fehler kapputt! Hat jedoch alles seine Ordnung, antwortet der CPC artig mit 'Danke'.

Beim Spielablauf achtet der CPC ordentlich darauf, ob nicht vielleicht geschummelt wird. Ist dies der Fall, erfolgt eine Ermahnung und eine Zeitstrafe.

Die gemessene Zeit wird nicht nur bis zum hundertsten Bruchteil einer Sekunde genau angezeigt, nein der CPC gibt auch seinen Kommentar dazu ('S U P E R' ... 'sehr schwach').

Programmtechnisch gesehen ein Klacks! Dafür aber vielleicht als 'Entschuldigung' für den sehr vernünftigen (!) Kauf des CPC für all die lieben Mitmenschen gedacht. Vielleicht machen Sie so auch noch Computerfreaks aus ihnen, dann ist die Freude und Begeisterung halt mindestens doppelt so groß wie zuvor. Noch ein Verwendungszweck dieses Programms: Messen Sie mal die Reaktionszeiten in verschiedenen Gemütslagen und Zuständen!

```
                    * * *  Z-80 Disassembler * * *

 Titel ? Einfache Addition in Maschinensprache

 Startadresse (in Dez.) ?
 Endadresse (in Dez.)   ?

 Datafelder vorhanden (J/ ) ?

 Drucker oder Bildschirm (D/B) ? b

                 Einfache Addition in Maschinensprache

 43776  2 Bytes      ABO  0          3E04      LD       A 04
 43778  2 Bytes      ABO  2          0607      LD       B 07
 43780  1 Bytes      ABO  4          80        ADD      A B
 43781  3 Bytes      ABO  5          32007D    LD        7D00  A
 43784  1 Bytes      ABO  8          C9        RET

* *  Ausdruck fertig  * *

Nochmal (J/ )
3280 DATA 3e,04,06,07,80,32,00,7d
Ready
█
```

```
10 REM Reaktion
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                              Rainer Lueers
40 REM Vorbedingungen treffen
50 INK 0,0:INK 1,24:PEN 1:PAPER 0:MODE 1
60 INPUT "Wie viele Teilnehmer (1 bis 4)
 ";f$
70 IF VAL(f$)<1 OR VAL(f$)>4 THEN GOTO 6
0
80 FOR n=1 TO VAL(f$)
90 PRINT:PRINT "Ihr Name, bitte, Spieler
";n;:INPUT f1$(n)
100 REM Reaktion auf falsche Eingabe
110 IF f1$(n)="" AND merker=0 THEN PRINT
:PRINT "Seien Sie doch nicht feige!
        Jede Sache hat auch einen Namen.
 Soweit ich dummer Computer weiss, gilt
das     auch fuer Menschen und Tiere!":m
erker=1:GOTO 90
120 IF f1$(n)="" AND merker=1 THEN PRINT
:PRINT "Diese Menschen, diese Menschen!
        Gehen wir also mal behutsam ans
Werk... vielleicht klappt es so! Ich hei
sse     CPC464. Nein, ich bin nicht vom
Mars... Mein Vorname ist Schneider (komi
sch,"
130 IF f1$(n)="" AND merker=1 THEN PRINT
 "wie?). Du darfst mich ruhig 'Schneider
' nennen ... aber nur wenn Du mir sagst,
  wie Du mit Vornamen heisst. Also bitte
!":merker=2:GOTO 90
140 IF f1$(n)="" AND merker=2 THEN PRINT
:PRINT "Noch eine Fehleingabe und der Co
mputer  ist kapputt ... ich darf doch bi
tten ...";:merker=3:GOTO 90
150 IF f1$(n)="" AND merker=3 THEN FOR n
=1 TO 10000:PRINT CHR$(PEEK(n));:NEXT
160 merker=0
170 NEXT n
```

```
18Ø MODE Ø:LOCATE 8,12:PRINT "Danke"
19Ø FOR n=1 TO 2ØØØ:NEXT n
2ØØ MODE 1:FOR n=1 TO VAL(f$)
21Ø PRINT f1$(n);"!!!!":PRINT:PRINT:PRIN
T "Du bist dran!":PRINT "Druecke eine me
iner Tasten, wenn Du      bemerkt hast, d
ass sich die              Bildschirmfarbe
 geaendert hat."
22Ø REM Berechnung der Zeitspanne
23Ø FOR m=1 TO 2ØØØ:NEXT m:a=1ØØØ*RND(1)
:FOR m=1 TO a:a$=INKEY$:IF a$="" THEN NE
XT m:GOSUB 28Ø ELSE MODE Ø:PRINT "Schumm
ler!!!!":PRINT "Du hast 5 Sekunde":PRINT
"gebraucht.":f(n)=5:FOR m=1 TO 2ØØØ:NEXT
 m:MODE 1
24Ø REM Endergebnis ausgeben
25Ø NEXT n:CLS:FOR n=1 TO VAL(f$):PRINT
f1$(n);" hat":PRINT f(n);"Sekunden gebra
ucht!":PRINT:NEXT n:PRINT:PRINT:INPUT "N
och ein Spiel ( /N) ";z$:IF LEFT$(UPPER$
(z$),1)="N" THEN END
26Ø PRINT:INPUT "Mit den gleichen Spiele
rn (J/ ) ";z$:IF LEFT$(UPPER$(z$),1)="J"
 THEN GOTO 2ØØ ELSE RUN
27Ø REM Unterprogramm zur Zeitkontrolle
28Ø MODE Ø:PAPER 1:CLS:b=TIME
29Ø a$=INKEY$:IF a$="" THEN GOTO 29Ø
3ØØ c=TIME:PAPER Ø:MODE 1:f(n)=(c-b)/25Ø
:PRINT f1$(n):PRINT "Du hast";f(n);"Seku
nden gebraucht!"
31Ø PRINT
32Ø REM Wertung
33Ø IF f(n)<Ø.ØØ5 THEN PRINT "S U P E R"
:GOTO 37Ø
34Ø IF f(n)<Ø.1 THEN PRINT "Ganz gut":GO
TO 37Ø
35Ø IF f(n)<Ø.5 THEN PRINT "schwach":GOT
O 37Ø
36Ø PRINT "Sehr, sehr schwach!"
37Ø FOR m=1 TO 2ØØØ:NEXT m:CLS:RETURN
```

Zahlsystemumrechner

Der CPC kann nicht nur Zahlen in dem uns vertrauten Dezimalsystem abbilden, er hat auch spezielle Befehle zum Umrechnen in andere Zahlsysteme so z.B. bedeutet '&H' einer Zahl vorangestellt: Hexadezimalzahl; '&X' bedeutet Binär- bzw. Dualzahl.

Auf diese Art und Weise können wir 'PRINT &HFF' eingeben und bekommen als Ergebnis dieser Hexadezimalzahl umgerechnet in eine Dezimalzahl '255' auf dem Bildschirm ausgegeben.

Geben wir 'PRINT &X111' ein, so erhalten wir gar als Ergebnis dieser Binärzahl umgerechnet in eine Dezimalzahl: '7'.

Schwierig wird es schon, wenn man direkt von einer Hexadezimalzahl in eine Binärzahl umrechnen will. Unmöglich wird es gar, wenn wir mit einem anderen geläufigen Zahlsystem außer hexadezimal, dezimal oder binär (z.B. oktal) arbeiten wollen.

Um die Zahlsysteme besser kennenzulernen ist nun dieses Programm entstanden. Sie können durch Voranstellung des entsprechenden Buchstabens ('H' für Hexadezimal, 'B' für Binaer, 'D' für Dezimal) jede eingegebene Zahl in jedem dieser drei Zahlsysteme darstellen lassen.

Außerdem können Sie durch Drücken der (ENTER)-Taste Ihre Dezimaleingaben in ein Zahlsystem zwischen 2 und 9 umrechnen. Hierbei geschieht etwas interessantes und gleichzeitig lehrreiches: Die Zahl in dem gewünschte Zahlsystem wird vor Ihren Augen ausgerechnet - Schritt für Schritt.

Wie geht das vor sich? Wir wählen ein Beispiel ... rechnen wir im Fünfer-Zahlensystem. Drücken Sie also (ENTER) auf die Frage nach 'Dez.,Hex. o. Bin.'. Auf die nächste Frage antworten Sie mit '5' und geben nun z.B. die Zahl '230' ein. Die ausgerechnete Zahl im Fünfersystem ist die '1410'.

Nun zur Erklärung des Vorgangs: Im Fünfer-Zahlsystem haben wi die Ziffern '0, 1, 2, 3 und 4'. Der CPC teilt unsere eingegeben Zahl '230' durch 5. Ergebnis: '46' Rest '0'. Anschließend wir die '46' geteilt: '9' Rest '1'. Nun wird die '9' geteilt: '1 Rest '4' und so bleibt schließlich ein Rest '4' übrig. Schreibe wir uns die Restwerte noch einmal auf:
1. 0 2. 1 3. 4. Beim letzten Wert nehmen wir nich nur den Rest '4', sondern auch die Anzahl = '1'.

So erhalten wir für Dezimal '230' die Zahl im Fünfersystem '1410'. Probe gefällig? Hierzu bilden wir Fünfer-Potenzen vo rechts nach links:
1) 0 * 5↑0 = 0 2) 1 * 5↑1 = 5 3) 4 * 5↑2 = 100
4) 1 * 5↑3 = 125 -) 0 + 5 + 100 + 125 = 230 !

Nur der Vollständigkeit halber erwähnt: Auch ohne das Program 'Zahlsystemumrechner' kann man mit dem CPC-Basic immerhin Dezi mal-Hexadezimal und Dezimal-Binär umrechnen und zwar mit de Befehlen: 'HEX$(' und 'BIN$('.

```
10 REM Zahlsystemumrechner
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effekt=2:
normal=1
50 ON ERROR GOTO 410
60 MODE 1
70 REM Auswahl zwischen den Zahlsystemen
80 INPUT "Eingabe (Dez.,Hex o. Bin.) ";a
$:IF VAL(a$)>65535 THEN GOTO 80 ELSE IF
a$="" THEN GOTO 160 ELSE b$=LEFT$(a$,1):
b$=UPPER$(b$):IF b$="H" THEN GOSUB 310 E
LSE IF b$="B" THEN GOSUB 360 ELSE IF b$=
"D" THEN a$=RIGHT$(a$,LEN(a$)-1)
90 REM Berechnung der Dezimal-, Binaer-
       und Hexadezimalzahlen fuer den
       Bildschirmausdruck
100 PRINT:PRINT "dezimal  binaer oder du
al   hexadezimal"
110 PRINT "------------------------------
----------"
120 PRINT a$;TAB(9);" ";
130 b$=BIN$(VAL(a$),16):PRINT LEFT$(b$,8
);" ";RIGHT$(b$,8);"  ";
140 PRINT TAB(32);HEX$(VAL(a$),4)
150 PRINT:PRINT:GOTO 80
160 MODE 1
170 REM Auswahl zwischen den
        Zahlsystemen 2 bis 9
180 INPUT "Welches Zahlsystem (2 bis 9)
";a
190 IF a=0 THEN RUN ELSE IF a<2 OR a>9 T
HEN GOSUB 440:GOTO 160
200 CLS:PRINT "Umrechnung DEZ. -";a;"ers
ystem":PRINT "==========================
==":PRINT
210 REM Berechnung der Zahlen im
        gewaehlten Zahlsystem (2 bis 9)
```

```
220 INPUT "Ihre Eingabe (DEZ.) ";b:IF b>
32767 THEN GOSUB 440:GOTO 160 ELSE PRINT
230 c=böa
240 PRINT "Anzahl:";c;
250 d=b MOD a
260 PRINT TAB(15);"MOD:";d
270 a$=RIGHT$(STR$(d),1)+a$
280 IF c<a THEN a$=STR$(c)+a$:PRINT:PRIN
T a$:a$="":PRINT:PRINT:PRINT:GOTO 180
290 b=c:GOTO 230
300 REM Eingabe der Hexadezimalzahl
        ueberpruefen und umrechnen
310 IF LEN(a$)>5 THEN RETURN
320 b$="&h "+MID$(a$,2,LEN(a$)-1)
330 IF VAL(b$)<0 THEN a$=STR$(VAL(b$)+65
536) ELSE a$=STR$(VAL(b$))
340 RETURN
350 REM Eingabe der Binaerzahl
        ueberpruefen und umrechnen
360 IF LEN(a$)>17 THEN RETURN
370 b$="&x "+MID$(a$,2,LEN(a$)-1)
380 IF VAL(b$)<0 THEN a$=STR$(VAL(b$)+65
536) ELSE a$=STR$(VAL(b$))
390 RETURN
400 REM Errorbehandlungsroutine
410 RESUME 420
420 RUN
430 END
440 PEN effekt:PRINT:PRINT TAB(12) "Fals
che Eingabe!"
450 PEN normal:GOSUB 460:RETURN
460 PRINT:PRINT TAB(7) "<Bitte eine Tast
e druecken>"
470 f$=INKEY$:IF f$="" THEN GOTO 470
480 RETURN
```

Disassembler

In den Programmen 'Speicher 1' bis 'Speicher 5' sowie im Variablen-Referenzlistenprogramm haben wir uns bereits ein klein wenig mit den 'Innereien' unseres CPC beschäftigt. Unter 'Innereien' sei hier verstanden: Wie geht es überhaupt vor sich, daß ein unmenschliches Wesen wie der Computer zu denken fähig ist (merken Sie sich dazu folgendes: Ein Computer ist immer nur so schlau, wie die Menschen, die ihn programmiert haben).

In den Programmen 'Speicher 1' bis 'Speicher 5' sind wir aber noch nicht tief genug in unseren CPC eingedrungen. Wir haben uns seinerzeit nämlich nur angeschaut, wie unsere BASIC-Programme Zeilennummer für Zeilennummer, Befehl für Befehl, im Speicher abgelegt werden, aber nichts über die Verarbeitung zu den Befehlen und zu dieser Art der Abspeicherung hin kennengelernt. Dabei wurde auch erwähnt, daß der CPC wie jeder andere Computer auch auf die Grundststellung zurückgelangt: an oder aus bzw. in binärer Form: 0 oder 1. Aus dem Erkennen einer Vielzahl von aufeinanderfolgenden 0 en oder 1 en erkennt der Computer dann ein Zeichen, manchmal auch einen Befehl.

Unser BASIC nennt sich eine höhere Programmiersprache: sie ist schon mit einer Vielzahl verständlicher Begriffe aufgebaut, so daß wir bei der Benutzung dieser Befehle vielmehr eine Vorstellung davon bekommen, was der Computer tun kann, als wenn wir nur stur 0 en oder 1 en in den Speicher eingeben würden.

Der in diesem Programm untersuchte Assemblercode ist eine Stufe zwischen BASIC und Maschinensprache. Wir haben hier eine Vielzahl von Befehlen zur Verfügung (beim Prozessor Z80A im CPC sind es gar mehr als 600), die allerdings bei weitem nicht so wirkungsvoll arbeiten wie die BASIC-Befehle. Im Speicher benötigen diese Befehle teils 1 Byte, teils 2 Bytes, manchmal auch gar 3 Bytes (1 Byte = 1 Speicherplatz). Das Programm 'Disassembler' hat sämtliche Z80A-Assemblercodes (man nennt sie auch Mnemoniks) gespeichert und untersucht den von uns vorgegebenen Speicherraum nach deren Vorkommen. Dabei untersucht es das Vorkommen bestimmter Zeichenfolgen (ähnlich wie bei dem 'Speicher'-Programm,

wo wir nach der Zeichenfolge 'Laenge' gesucht haben); werden diese Zahlenfolgen irgendwo angetroffen, wird der entsprechende Assemblercode auf den Bildschirm (oder Drucker) geschrieben.

Da wir nicht davon ausgehen können, daß Sie Tag für Tag sehr viele Maschinenprogramme schreiben, haben wir einen Lerneffekt in unser 'Disassembler'-Proramm eingebaut: In Zeile 3530 steht eine DATA-Anweisung, gefolgt von meheren Hexadezimalzahlen. Diese Zahlen sind ein kleines Maschinenprogramm, in dem zwei Zahlen zusammengezählt und an einer bestimmten Speicherstelle abgelegt werden. Sie können sich die Umwandlung dieser Zahlen in ein Assemblerprogramm anschauen, indem Sie nach dem Programmstart auf alle Fragen ohne weitere Eingabe nur mit (ENTER) antworten.

Sie können nun auch andere Zahlen (in hexadezimaler Form, bitte!) in die DATA-Zeile schreiben - oder Sie finden ein längeres Maschinenprogramm für Z80A-Prozessor in einer Computerzeitschrift ... Sie können auch ruhig mehrere DATA-Zeilen mit Zahlen anfüllen ... Viel Spaß dabei!

```
10 REM Disassembler
20 REM CPC464 Basic Programme
30 REM Copyright 1984 DATA BECKER &
                          Rainer Lueers
40 CLEAR
50 REM Speicherplatz ab 43776 fuer
       Maschinenroutinen frei halten
60 MEMORY 43775:GOSUB 3450
70 DIM i$(255),bl(255):n=0:m=0:sy=1
80 GOTO 1280
90 MODE 2:c=0:PRINT TAB(20)"* * *  Z-80
Disassembler * * *"
100 REM Wuensche des Benutzers
        abklaeren
110 PRINT:INPUT "Titel ";f$
120 IF LEN(f$)>80 THEN GOTO 110
130 PRINT
140 REM Anfangsadresse der
        abgespeicherten Maschinenroutine
        in DATA-Zeilen
150 INPUT "Startadresse (in Dez.) ";a:IF
 a<0 OR a>65535 THEN GOTO 130 ELSE IF a=
0 THEN a=43776
160 INPUT "Endadresse (in Dez.)   ";b
170 IF b<e OR b>65535 OR b<0 THEN GOTO 1
60 ELSE IF b=0 THEN b=zaehler1
180 PRINT:INPUT "Datafelder vorhanden (J
/ ) ";v$
190 v$=LEFT$(UPPER$(v$),1):IF v$="J" THE
N GOSUB 2140
200 PRINT:INPUT "Drucker oder Bildschirm
 (D/B) ";b$
210 REM Festlegen der Variable 'drubi',
        die fuer die Ansprache des
        PRINT-Befehls zustaendig ist:
        8 -> Drucker, 0 -> Bildschirm
220 b$=LEFT$(UPPER$(b$),1):IF b$="D" THE
N drubi=8 ELSE drubi=0
230 PRINT #drubi
```

```
240 REM Ueberschrift = Titel, der durch
         die Formel in der Mitte der
         Zeile erscheint
250 PRINT #drubi,TAB((80-LEN(f$))/2-5)f$
260 PRINT #drubi
270 REM Beginn des eigentlichen
         Disassemblers. Sukksessives
         Einlesen des angegebenen
         Speicherbereichs und
         Interpretieren der Daten
         in Maschinenbefehle=Mnemoniks
280 q=a
290 h1=INT(a/4096):h2=INT((a-h1*4096)/25
6)
300 l1=INT((a-h1*4096-h2*256)/16)
310 l2=a-h1*4096-h2*256-l1*16
320 IF m<>0 THEN GOTO 340
330 GOTO 390
340 IF a>=a(n) AND a<=b(n) THEN GOTO 360
350 GOTO 390
360 d=PEEK(a):GOSUB 3380:f$="D":fl$="A":
gh$="T":gl$="A":t$=i$(d):i$(d)=""
370 IF a=b(n) THEN n=n+1:m=m-1:GOTO 400
380 GOTO 400
390 d=PEEK(a):GOSUB 650
400 v=d:GOSUB 2740
410 dh$=h$:dl$=l$
420 IF h1>=10 THEN GOTO 560
430 h1$=STR$(h1):h1$=MID$(h1$,2,1)
440 IF h2>=10 THEN GOTO 570
450 h2$=STR$(h2):h2$=MID$(h2$,2,1)
460 IF l1>=10 THEN GOTO 580
470 l1$=STR$(l1):l1$=MID$(l1$,2,1)
480 IF l2>=10 THEN GOTO 590
490 l2$=STR$(l2):l2$=MID$(l2$,2,1)
500 REM Ausgabe des disassemblierten
         Speicherbereichs
```

```
510 qt=qt+1:IF qt>105 THEN qt=0:FOR qi=1
 TO 9:PRINT #drubi:NEXT qi:GOSUB 3400:sy
=sy+1:PRINT #drubi,TAB(30) "Seite";sy:PR
INT #drubi:GOTO 510 ELSE PRINT #drubi,q;
LEN(dh$+dl$+eh$+el$+fh$+fl$+gh$+gl$)/2;"
Bytes";TAB(18);
520 PRINT #drubi,USING "    !!!!";" ",h1$
,h2$,l1$,l2$;:PRINT #drubi,"            "
;:PRINT #drubi,USING "!!!!!!!!";" ",dh$,
dl$,eh$,el$,fh$,fl$,gh$,gl$;:PRINT #drub
i,TAB(50);i$(d)
530 a=a+1:c=c+1:IF a>=b+1 THEN GOTO 600
540 IF gh$="T" THEN i$(d)=t$
550 GOTO 280
560 x=h1-10+65:h1$=CHR$(x):GOTO 440
570 x=h2-10+65:h2$=CHR$(x):GOTO 460
580 x=l1-10+65:l1$=CHR$(x):GOTO 480
590 x=l2-10+65:l2$=CHR$(x):GOTO 510
600 PRINT:PRINT "* *  Ausdruck fertig  *
 *"
610 REM Frage nach eventuellem
        Neustart
620 PRINT:INPUT "Nochmal (J/ ) ";b$:b$=L
EFT$(UPPER$(b$),1)
630 IF b$="J" THEN RUN
640 END
650 IF d<64 OR d>127 THEN GOTO 710
660 IF d=118 THEN RETURN
670 dh=INT(d/16):dl=d-dh*16
680 g=dl AND 7:f=((d AND 56)/8)
690 i$(d)="LD         "+j$(f)+","
700 i$(d)=i$(d)+j$(g):GOSUB 3380:RETURN
710 IF d<128 OR d>191 THEN GOTO 830
720 dh=INT(d/16):dl=d-(dh*16)
730 g=dl AND 7:f=((d AND 120)/8)
740 IF f=0 THEN i$(d)="ADD       A,"
750 IF f=1 THEN i$(d)="ADC       A,"
760 IF f=2 THEN i$(d)="SUB       "
770 IF f=3 THEN i$(d)="SBC       A,"
780 IF f=4 THEN i$(d)="AND       "
```

```
790 IF f=5 THEN i$(d)="XOR        "
800 IF f=6 THEN i$(d)="OR         "
810 IF f=7 THEN i$(d)="CP         "
820 i$(d)=i$(d)+j$(g):GOSUB 3380:RETURN
830 IF bl(d)=1 THEN GOTO 880
840 IF bl(d)=2 THEN GOTO 890
850 IF bl(d)=3 THEN GOTO 900
860 IF bl(d)=4 THEN GOTO 1770
870 IF d=221 OR d=253 THEN GOTO 2220
880 GOSUB 3380:RETURN
890 a=a+1:z=PEEK(a):GOTO 910
900 a=a+1:z=PEEK(a):a=a+1:z1=PEEK(a)
910 v=z:GOSUB 2740:eh$=h$:el$=l$
920 IF bl(d)=3 THEN GOTO 940
930 GOSUB 3390:GOTO 950
940 v=z1:GOSUB 2740:fh$=h$:fl$=l$
950 p1$=fh$+fl$+eh$+el$:p2$=eh$+el$:gh$=
"":gl$=""
960 f=d AND 7:g=d AND 56:g=g/8
970 IF f=6 AND (d AND 192)=0 THEN GOTO 1
230
980 IF f=2 AND (d AND 192)=192 THEN GOTO
1260
990 IF f=4 THEN GOTO 1270
1000 IF f=0 AND (g<>2) THEN GOTO 1240
1010 i$(33)="LD          HL,"+p1$
1020 i$(34)="LD          ("+p1$+"),HL"
1030 i$(50)="LD          ("+p1$+"),A"
1040 i$(205)="CALL        "+p1$
1050 i$(195)="JP          "+p1$
1060 i$(58)="LD          A,("+p1$+")"
1070 i$(254)="CP          "+p2$
1080 i$(42)="LD          HL,("+p1$+")"
1090 i$(49)="LD          SP,"+p1$
1100 i$(17)="LD          DE,"+p1$
1110 i$(16)="DJNZ        "+p2$
1120 i$(1)="LD          BC,"+p1$
1130 i$(198)="ADD         A,"+p2$
1140 i$(206)="ADC         A,"+p2$
1150 i$(211)="OUT         "+p2$+",A"
```

```
1160 i$(214)="SUB          "+p2$
1170 i$(219)="IN           A,"+p2$
1180 i$(222)="SBC          A,"+p2$
1190 i$(230)="AND          "+p2$
1200 i$(238)="XOR          "+p2$
1210 i$(246)="OR           "+p2$
1220 RETURN
1230 i$(d)="LD          "+j$(g)+","+p2$:RE
TURN
1240 IF g=3 THEN i$(d)="JR          "+p2$:
RETURN
1250 g=g-4:i$(d)="JR          "+s$(g)+","+
p2$:RETURN
1260 i$(d)="JP          "+s$(g)+","+p1$:RE
TURN
1270 i$(d)="CALL        "+s$(g)+","+p1$:RE
TURN
1280 FOR s=0 TO 63:bl(s)=1:NEXT s
1290 FOR s=192 TO 255:bl(s)=1:NEXT s
1300 bl(118)=1:bl(6)=2:bl(14)=2:bl(16)=2
:bl(22)=2:bl(24)=2:bl(30)=2:bl(32)=2:bl(
38)=2:bl(40)=2:bl(46)=2:bl(48)=2:bl(54)=
2:bl(56)=2
1310 bl(62)=2:bl(198)=2:bl(206)=2:bl(211
)=2:bl(214)=2:bl(219)=2:bl(222)=2:bl(230
)=2:bl(238)=2:bl(246)=2:bl(254)=2
1320 bl(1)=3:bl(17)=3:bl(33)=3:bl(34)=3:
bl(42)=3:bl(49)=3:bl(50)=3:bl(58)=3:bl(1
94)=3:bl(195)=3:bl(196)=3:bl(202)=3:bl(2
04)=3:bl(205)=3:bl(210)=3:bl(212)=3:bl(2
18)=3:bl(220)=3
1330 bl(226)=3:bl(228)=3:bl(234)=3:bl(23
6)=3:bl(242)=3:bl(244)=3:bl(250)=3:bl(25
2)=3:bl(203)=3:bl(237)=3:bl(221)=0:bl(25
3)=3
1340 j$(0)="B":j$(1)="C":j$(2)="D":j$(3)
="E":j$(4)="H":j$(5)="L"
1350 j$(6)="(HL)":j$(7)="A"
```

```
1360 s$(Ø)="NZ":s$(1)="Z":s$(2)="NC":s$(
3)="C":s$(4)="PO":s$(5)="PE":s$(6)="P":s
$(7)="M"
1370 p$(Ø)="SBC        HL,":p$(1)="ADC
    HL,"
1380 p$(2)="SBC        HL,":p$(3)="ADC
    HL,"
1390 p$(4)="SBC        HL,":p$(5)="ADC
    HL,"
1400 p$(7)="ADC        HL,":n$(Ø)="BC":n$
(1)="BC":n$(2)="DE"
1410 n$(3)="DE":n$(4)="HL":n$(5)="HL":n$
(7)="SP"
1420 l$(Ø)="N":l$(1)="I":m$(Ø)="Ø":m$(2)
="1":m$(3)="2"
1430 o$(Ø)="I,A":o$(1)="R,A":o$(2)="A,I"
:o$(3)="A,R"
1440 q$(Ø)="LD":q$(1)="CP":q$(2)="IN":q$
(3)="OUT"
1450 i$(Ø)="NOP":i$(2)="LD          (BC),A
":i$(3)="INC          BC"
1460 i$(7)="RLC        A":i$(8)="EX
  AF,AF'"
1470 i$(9)="ADD        HL,BC":i$(10)="LD
       A,(BC)"
1480 i$(11)="DEC        BC":i$(15)="RRC
     A"
1490 i$(18)="LD         (DE),A":i$(19)="I
NC        DE"
1500 i$(23)="RLA":i$(25)="ADD        HL,D
E":i$(26)="LD        A,(DE)"
1510 i$(27)="DEC        DE":i$(31)="RRA":
i$(35)="INC        HL"
1520 i$(39)="DAA":i$(41)="ADD        HL,H
L":i$(43)="DEC        HL"
1530 i$(47)="CPL":i$(51)="INC        SP":
i$(55)="SCF"
1540 i$(57)="ADD        HL,SP":i$(59)="DE
C        SP":i$(63)="CCF"
1550 i$(43)="DEC        HL"
```

```
1560 i$(192)="RET         NZ":i$(197)="PUS
H       BC"
1570 i$(199)="RST         Ø":i$(200)="RET
      Z":i$(207)="RST       8"
1580 i$(208)="RET         NC":i$(209)="POP
       DE"
1590 i$(213)="PUSH        DE":i$(215)="RST
       1ØH"
1600 i$(216)="RET         C":i$(223)="RST
      18H"
1610 i$(224)="RET         PO":i$(227)="EX
       (SP),HL"
1620 i$(231)="RST         2ØH":i$(232)="RE
T        PE"
1630 i$(233)="JP          (HL)":i$(235)="E
X          DE,HL"
1640 i$(239)="RST         28H":i$(240)="RE
T        P"
1650 i$(201)="RET"
1660 i$(241)="POP         AF":i$(243)="DI"
:i$(245)="PUSH        AF"
1670 i$(247)="RST         3ØH":i$(248)="RE
T        M":i$(251)="EI"
1680 i$(249)="LD          SP,HL":i$(255)="
RST         38H"
1690 i$(197)="PUSH        BC"
1700 i$(193)="POP         BC":i$(229)="PUS
H       HL"
1710 i$(225)="POP         HL"
1720 i$(217)="EXX"
1730 i$(118)="HALT"
1740 l=Ø:FOR d=5 TO 45 STEP 8:i$(d)="DEC
       "+j$(l):l=l+1:NEXT d
1750 l=Ø:FOR d=4 TO 44 STEP 8:i$(d)="INC
       "+j$(l):l=l+1:NEXT d
1760 i$(6Ø)="INC        A":i$(61)="DEC
    A":GOTO 9Ø
1770 IF d=2Ø3 THEN GOTO 179Ø
1780 IF d=237 THEN GOTO 295Ø
1790 a=a+1:z=PEEK(a)
```

```
1800 eh=INT(z/16):el=z-eh*16:g=el AND 7
1810 f=INT((el AND 8)/8):f=f+(eh*2):GOSU
B 1830
1820 GOTO 1980
1830 IF f=0 THEN i$(d)="RLC        ":RETU
RN
1840 IF f=1 THEN i$(d)="RRC        ":RETU
RN
1850 IF f=2 THEN i$(d)="RL         ":RETU
RN
1860 IF f=3 THEN i$(d)="RR         ":RETU
RN
1870 IF f=4 THEN i$(d)="SLA        ":RETU
RN
1880 IF f=5 THEN i$(d)="SRA        ":RETU
RN
1890 IF f=7 THEN i$(d)="SRL        ":RETU
RN
1900 IF f>=8 AND f<=15 THEN GOTO 1940
1910 IF f>=16 AND f<=23 THEN GOTO 1960
1920 f=f AND 7:f$=CHR$(f+48)
1930 i$(d)="SET        "+f$+",":RETURN
1940 f=f AND 7:f$=CHR$(f+48)
1950 i$(d)="BIT        "+f$+",":RETURN
1960 f=f AND 7:f$=CHR$(f+48)
1970 i$(d)="RES        "+f$+",":RETURN
1980 IF eh>=10 THEN GOTO 2030
1990 eh$=STR$(eh):eh$=MID$(eh$,2,1)
2000 IF el>=10 THEN GOTO 2040
2010 el$=STR$(el):el$=MID$(el$,2,1)
2020 GOSUB 3390:i$(d)=i$(d)+j$(g):RETURN
2030 x=eh+55:eh$=CHR$(x):GOTO 2000
2040 x=el+55:el$=CHR$(x):GOTO 2020
2050 IF d AND 7=5 THEN GOTO 2080
2060 IF d AND 7=4 THEN GOTO 2080
2070 GOTO 880
2080 l=d AND 56
2090 g=l/8
2100 IF d AND 7=5 THEN i$(d)="DEC
"+j$(g):GOTO 2120
```

```
2110 i$(d)="INC         "+j$(g)
2120 GOSUB 3380:RETURN
2130 REM Unterprogramm zum Ausschliessen
         des Disassemblierens von
         bestimmten Speicherbereichen
2140 n=1
2150 INPUT "Startadr.=";a(n)
2160 INPUT "Endadr.=";b(n)
2170 IF a(n)<a OR b(n)>b THEN GOTO 2150
2180 INPUT "Noch ein Teil (J/ ) ";v$
2190 v$=LEFT$(UPPER$(v$),1)
2200 IF v$="J" THEN n=n+1:GOTO 2150
2210 m=n:n=1:RETURN
2220 IF d=221 THEN v$="IX":GOTO 2240
2230 v$="IY"
2240 a=a+1:z=PEEK(a)
2250 IF z=203 THEN GOTO 2660
2260 IF z>=70 AND z<=190 THEN GOTO 2810
2270 IF z=33 OR z=34 OR z=42 THEN GOTO 2
430
2280 IF z=52 OR z=53 THEN GOTO 2530
2290 IF z=54 THEN GOTO 2590
2300 v=z:GOSUB 2740:eh$=h$:el$=l$:GOSUB
3390
2310 IF z=9 THEN i$(d)="ADD         "+v$+
",BC"
2320 IF z=25 THEN i$(d)="ADD        "+v$+
",DE"
2330 IF z=35 THEN i$(d)="INC        "+v$
2340 IF z=41 THEN i$(d)="ADD        "+v$+
","+v$
2350 IF z=43 THEN i$(d)="DEC        "+v$
2360 IF z=57 THEN i$(d)="ADD        "+v$+
",SP"
2370 IF z=225 THEN i$(d)="POP        "+v$
2380 IF z=227 THEN i$(d)="EX        (SP)
,"+v$
2390 IF z=229 THEN i$(d)="PUSH       "+v$
2400 IF z=233 THEN i$(d)="JP        ("+v
$+")"
```

```
2410 IF z=249 THEN i$(d)="LD         SP,"
+v$
2420 RETURN
2430 v=z:GOSUB 2740
2440 eh$=h$:el$=l$
2450 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2460 fh$=h$:fl$=l$:a=a+1
2470 z1=PEEK(a):v=z1:GOSUB 2740
2480 gh$=h$:gl$=l$
2490 IF z=33 THEN i$(d)="LD         "+v$+
","+gh$+gl$+fh$+fl$
2500 IF z=34 THEN i$(d)="LD         ("+gh
$+gl$+fh$+fl$+"),"+v$
2510 IF z=42 THEN i$(d)="LD         "+v$+
",("+gh$+gl$+fh$+fl$+")"
2520 RETURN
2530 v=z:GOSUB 2740:eh$=h$:el$=l$
2540 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2550 fh$=h$:fl$=l$:gh$="":gl$=""
2560 IF z=52 THEN i$(d)="INC        ("+v
$+"+"+fh$+fl$+")"
2570 IF z=53 THEN i$(d)="DEC        ("+v
$+"+"+fh$+fl$+")"
2580 RETURN
2590 v=z:GOSUB 2740:eh$=h$:el$=l$
2600 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2610 fh$=h$:fl$=l$:a=a+1
2620 z1=PEEK(a):v=z1:GOSUB 2740
2630 gh$=h$:gl$=l$
2640 i$(d)="LD         ("+v$+"+"+fh$+fl$+
"),"+gh$+gl$
2650 RETURN
2660 a=a+1:eh$=CHR$(67):el$=CHR$(66)
2670 v=PEEK(a):GOSUB 2740
2680 fh$=h$:fl$=l$
2690 a=a+1:z2=PEEK(a):o=z2 AND 248:o=o/8
2700 q$="("+v$+"+"+fh$+fl$+")"
2710 f=o:GOSUB 1830
2720 i$(d)=i$(d)+q$
```

```
2730 v=z2:GOSUB 2740:gh$=h$:gl$=l$:RETUR
N
2740 h=INT(v/16):l=v-(h*16)
2750 IF h>=10 GOTO 2790
2760 h$=STR$(h):h$=MID$(h$,2,1)
2770 IF l>=10 THEN GOTO 2800
2780 l$=STR$(l):l$=MID$(l$,2,1):RETURN
2790 x1=h+55:h$=CHR$(x1):GOTO 2770
2800 x1=l+55:l$=CHR$(x1):RETURN
2810 v=z:GOSUB 2740:eh$=h$:el$=l$:gh$=""
:gl$=""
2820 a=a+1:z1=PEEK(a)
2830 v=z1:GOSUB 2740:fh$=h$:fl$=l$
2840 IF z=126 THEN i$(d)="LD         A,("
+v$+"+"+fh$+fl$+")":RETURN
2850 p=z AND 240
2860 IF p=112 THEN GOTO 2900
2870 IF p>=128 THEN GOTO 2920
2880 p=z AND 56:p=p/8:GOSUB 3290
2890 i$(d)="LD         "+g$+",("+v$+"+"+f
h$+fl$+")":RETURN
2900 p=z AND 7:GOSUB 3290
2910 i$(d)="LD         ("+v$+"+"+fh$+fl$+
"),"+g$:RETURN
2920 p=z AND 56:p=p/8:GOSUB 3290
2930 i$(d)=i$(d)+v$+"+"+fh$+fl$+")"
2940 RETURN
2950 a=a+1:z=PEEK(a):v=z:GOSUB 2740:eh$=
h$:el$=l$
2960 IF z=67 OR z=75 OR z=83 OR z=91 OR
z=115 OR z=123 THEN GOTO 3200
2970 GOSUB 3390:f=z AND 248:g=z AND 7
2980 IF f=160 THEN GOTO 3160
2990 IF f=168 THEN GOTO 3170
3000 IF f=176 THEN GOTO 3180
3010 IF f=184 THEN GOTO 3190
3020 f=z AND 56:f=f/8:g=z AND 7
3030 IF f=6 THEN i$(d)="SBC        HL,SP"
:RETURN
```

```
3040 IF g=0 THEN i$(d)="IN        "+j$(f
)+",(C)":RETURN
3050 IF g=1 THEN i$(d)="OUT       (C),"+
j$(f):RETURN
3060 IF g=2 THEN i$(d)=p$(f)+n$(f):RETUR
N
3070 IF g=4 THEN i$(d)="NEG":RETURN
3080 IF g=5 THEN i$(d)="RET"+l$(f):RETUR
N
3090 IF g=6 THEN i$(d)="IM "+m$(f):RETUR
N
3100 IF g<>7 THEN i$(d)="* * *":RETURN
3110 IF f<=3 THEN GOTO 3150
3120 IF f=4 THEN i$(d)="RRD":RETURN
3130 IF f=5 THEN i$(d)="RLD":RETURN
3140 GOTO 3100
3150 i$(d)="LD        "+o$(f):RETURN
3160 i$(d)=q$(g)+"I":RETURN
3170 i$(d)=q$(g)+"D":RETURN
3180 i$(d)=q$(g)+"IR":RETURN
3190 i$(d)=q$(g)+"DR":RETURN
3200 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740:fh
$=h$:fl$=l$
3210 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740:gh
$=h$:gl$=l$
3220 gg$=gh$+gl$+fh$+fl$
3230 IF z=67 THEN i$(d)="LD       ("+gg
$+"),BC":RETURN
3240 IF z=75 THEN i$(d)="LD       BC,("
+gg$+")":RETURN
3250 IF z=83 THEN i$(d)="LD       ("+gg
$+"),DE":RETURN
3260 IF z=91 THEN i$(d)="LD       DE,("
+gg$+")":RETURN
3270 IF z=115 THEN i$(d)="LD       ("+g
g$+"),SP":RETURN
3280 IF z=123 THEN i$(d)="LD       SP,(
"+gg$+")":RETURN
3290 IF p=0 THEN g$="B":i$(d)="ADD
  A,("
```

```
3300 IF p=1 THEN g$="C":i$(d)="ADC
A,("
3310 IF p=2 THEN g$="D":i$(d)="SUB        (
"
3320 IF p=3 THEN g$="E":i$(d)="SBC        A
,("
3330 IF p=4 THEN g$="H":i$(d)="AND        (
"
3340 IF p=5 THEN g$="L":i$(d)="XOR        (
"
3350 IF p=6 THEN g$="***":i$(d)="OR
 ("
3360 IF p=7 THEN g$="A":i$(d)="CP         (
"
3370 RETURN
3380 eh$="":el$="":fh$="":fl$="":gh$="":
gl$="":RETURN
3390 fh$="":fl$="":gh$="":gl$="":RETURN
3400 PRINT #drubi:RETURN
3410 PRINT #drubi:qt=0:sy=1
3420 PRINT #drubi:RETURN
3430 REM Ende der
        Unterprogrammbibliothek zur
        Erzeugung des Assemblercodes
3440 REM Ueberpruefung der Anzahl der zu
        disassemblierenden Bytes
        (hexadezimale Eingabe) in den
        DATA-Zeilen
3450 ON ERROR GOTO 3540:FOR zaehler1=437
76 TO 65535:READ a$:NEXT zaehler1
3460 REM Eigentliches Einlesen der
        Bytes in der DATA-Zeile in
        den Speicher ab 43776
3470 RESTORE:FOR m=43776 TO zaehler1-1
3480 READ a$
3490 a=VAL("&h"+a$)
3500 POKE m,a
3510 NEXT m:POKE m,&C9
3520 RETURN
3530 DATA 3e,04,06,07,80,32,00,7d
3540 RESUME 3470
```

Die Tokens des CPC464 (siehe 'Speicher 1' bis 'Speicher 5')
----

32 bis 126: normaler ASCII-Zeichensatz (s.Handbuch An.III,S.1)

128	AFTER	129	AUTO	130	BORDER	131	CALL
132	CAT	133	CHAIN	134	CLEAR	135	CLG
136	CLOSEIN	137	CLOSEOUT	138	CLS	139	CONT
140	DATA	141	DEF	142	DEFINT	143	DEFREAL
144	DEFSTR	145	DEG	146	DELETE	147	DIM
148	DRAW	149	DRAWR	150	EDIT	151	ELSE
152	END	153	ENT	154	ENV	155	ERASE
156	ERROR	157	EVERY	158	FOR	159	GOSUB
160	GOTO	161	IF	162	INK	163	INPUT
164	KEY	165	LET	166	LINE	167	LIST
168	LOAD	169	LOCATE	170	MEMORY	171	MERGE
172	MID$	173	MODE	174	MOVE	175	MOVER
176	NEXT	177	NEW	178	ON	179	ON BREAK
180	ON ERROR GOTO 0			181	ON SQ	182	OPENIN
183	OPENOUT	184	ORIGIN	185	OUT	186	PAPER
187	PEN	188	PLOT	189	PLOTR	190	POKE
191	PRINT	192	' (nicht ASCII, sondern Befehl 'REM')				
193	RAD	194	RANDOMIZE	195	READ	196	RELEASE
197	REM	198	RENUM	199	RESTORE	200	RESUME
201	RETURN	202	RUN	203	SAVE	204	SOUND
205	SPEED	206	STOP	207	SYMBOL	208	TAG
209	TAGOFF	210	TROFF	211	TRON	212	WAIT
213	WEND	214	WHILE	215	WIDTH	216	WINDOW
217	WRITE	218	ZONE	219	DI	220	EI
221 bis 226: keine direkt sichtbaren Tokens							
227	ERL	228	FN	229	SPC	230	STEP
231	SWAP	232 und 233: siehe 221				234	TAB
235	THEN	236	TO	237	USING	238	> (Rechnung)
239	= (Rech.)	240	>= (Re.)	241	< (Rech.)	242	<> (Rechnung)
243	<= (Re.)	244	+ (Re.)	245	- (Rech.)	246	* (Rechnung)
247	/ (Rech.)	248	^ (Re.)	249	\ (Rech.)	250	AND
251	MOD	252	OR	253	XOR	254	NOT

Mit dem Vorzeichen '255' entstehen folgende Befehlswörter:

0	ABS	1	ASC	2	ATN	3	CHR$
4	CINT	5	COS	6	CREAL	7	EXP
8	FIX	9	FRE	10	INKEY	11	INP
12	INT	13	JOY	14	LEN	15	LOG
16	LOG10	17	LOWER$	18	PEEK	19	REMAIN
20	SGN	21	SIN	22	SPACE$	23	SQ
24	SQR	25	STR$	26	TAN	27	UNT
28	UPPER$	29	VAL				
30 bis 63: keine direkt sichtbaren Tokens							
64	EOF	65	ERR	66	HIMEM	67	INKEY$
68	PI	69	RND	70	TIME	71	XPOS
72	YPOS	73 bis 112: keine direkt sichtaren Tokens					
113	BIN$	114	DEC$	115	HEX$	116	INSTR
117	LEFT$	118	MAX	119	MIN	120	POS
121	RIGHT$	122	ROUND	123	STRING$	124	TEST
125	TESTR	126	nicht sichtbar			127	VPOS

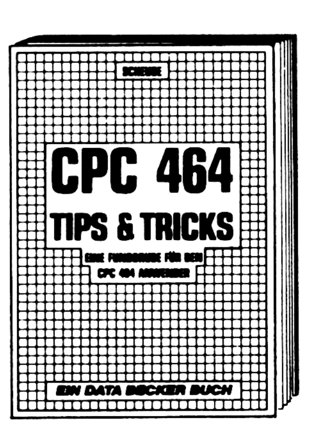
Die Neuen Bücher von DATA BECKER



Mit dem neuen DATA BECKER Einsteigerbuch den brandneuen CPC 464 kennenlernen.
Wer sich für den brandneuen Schneider-Homecomputer CPC 464 entschieden hat, findet mit dem DATA BECKER Buch „CPC 464 für Einsteiger" gleich den richtigen Start. Neben den wichtigsten Hinweisen über Handhabung und Anschlußmöglichkeiten bringt das Buch erste Hilfen für eigene Programme auf dem CPC 464. Zahlreiche Abbildungen und Bildschirmfotos ergänzen den Text. Das ideale Buch für jeden, der mit dem CPC 464 das Computern beginnen will.
CPC 464 FÜR EINSTEIGER, 1984,
über 200 Seiten, DM 29,–.



Der CPC 464 ist nicht nur zum Spielen da
Das neue Schulbuch zum CPC 464 von Professor Voß enthält, didaktisch gut aufbereitet, viele interessante Problemlösungs- und Lernprogramme (quadratische Gleichungen, exponentielles Wachstum, Geschichtszahlen, engl. Vokabeln lernen und vieles mehr). Dieses Buch ist nicht nur für Schüler bestens geeignet, sondern für jeden, der in die Programmierung wissenschaftlicher Probleme einsteigen will.
DAS SCHULBUCH ZUM CPC 464, 1984,
ca. 380 Seiten, DM 49,-



Viele Tips und Tricks rund um den CPC 464
Vom Hardwareaufbau, Betriebssystem, Basic-Tokens, Zeichnen mit dem Joystick, Anwendungen der Windowtechnologie und sehr vielen interessanten Programmen wie einer umfangreichen Dateiverwaltung, Soundeditor, komfortablen Zeichengenerator bis zu kompletten Listings spannender Spiele bietet das Buch viele Anregungen und wichtige Hilfen. Diese riesige Fundgrube sollte jeder CPC 464-Besitzer haben!
CPC 464 TIPS & TRICKS, 1984,
über 250 Seiten, DM 39,-

DAS STEHT DRIN:

Dieses Buch enthält Spitzenprogramme, vom Disassembler bis zum Sporttabellenprogramm. Mit spannenden Superspielen und kompletten Anwendungsprogrammen.

Aus dem Inhalt:

- Hexdump
- Grafikeditor
- Soundeditor
- Deutsche Umlaute
- Mathematikzeichensatz
- Ausführliche Fehlermeldungen
- Variablenreferenzliste
- Kalender
- Disassembler
- Langspielplattenverwaltung
- Texteditor
- Codeknacker
- Zahlsystemumrechner

UND GESCHRIEBEN HAT DIESES BUCH:

Rainer Lüers ist Computer-Fachtrainer bei einer großen Kaufhauskette und erfahrener Autor mehrerer EDV-Bücher.

ISBN 3-89011-049-5

Lüers / CPC 464 Basic-Programme

CPC 464



Document numérisé avec amour par :

AMSTRAD CPC

MÉMOIRE ÉCRITE

ACME

https://acpc.me/